NPS55-90-10

NAVAL POSTGRADUATE SCHOOL

Monterey, California

Nonlinear Modeling of Time Series using Multivariate Adaptive Regression Splines (MARS)

by Peter A. W. Lewis
and
James G. Stevens

April 1990

Prepared for:
Chief of Naval Research
Arlington, VA

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

Rear Admiral R. W. West, Jr.
Superintendent

Harrison Shull
Provost

This report was sponsored by the Chief of Naval Research and funded by the Naval Postgraduate School.

This report was prepared by:

REPORT DOCUMENTATION PAGE

1a Report Security Classification: Unclassified	1b Restrictive Markings
2a Security Classification Authority	3 Distribution Availability of Report: Approved for public release; distribution is unlimited.
2b Declassification/Downgrading Schedule	
4 Performing Organization Report Number(s): NPS55-90-10	5 Monitoring Organization Report Number(s)
6a Name of Performing Organization: Naval Postgraduate School; 6b Office Symbol (If Applicable): OR	7a Name of Monitoring Organization: Chief of Naval Research
6c Address (city, state, and ZIP code): Monterey, CA 93943-5000	7b Address (city, state, and ZIP code): Arlington, VA
8a Name of Funding/Sponsoring Organization: Naval Postgraduate School; 8b Office Symbol (If Applicable)	9 Procurement Instrument Identification Number: O&MN, Direct Funding
8c Address (city, state, and ZIP code)	10 Source of Funding Numbers: Program Element Number; Project No; Task No; Work Unit Accession No

11 Title (Include Security Classification)
Nonlinear Modeling of Time Series using Multivariate Adaptive Regression Splines (MARS)

12 Personal Author(s) Peter A. W. Lewis and James G. Stevens

13a Type of Report	13b Time Covered From To	14 Date of Report (year, month,day)	15 Page Count
Technical		1990, April	36

16 Supplementary Notation The views expressed in this paper are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes		
Field	Group	Subgroup

18 Subject Terms (continue on reverse if necessary and identify by block number)
Multivariate adaptive regression splines; nonlinear time series models; recursive partitioning; regression splines; threshold models; ASTAR models; Wolf sunspot numbers; limit cycles

19 Abstract (continue on reverse if necessary and identify by block number

MARS is a new methodology, due to Friedman, for nonlinear regression modeling. MARS can be conceptualized as a generalization of recursive partitioning that uses spline fitting in lieu of other simple functions. Given a set of predictor variables, MARS fits a model in a form of an expansion in product spline basis functions of predictors chosen during a forward and backward recursive partitioning strategy. MARS produces continuous models for discrete data that can have multiple partitions and multilinear terms. Predictor variable contributions and interactions in a MARS model may be analyzed using an ANOVA style decomposition.

By letting the predictor variables in MARS be lagged values of a time series, one obtains a new method for nonlinear autoregressive threshold modeling of time series. A significant feature of this extension of MARS is its ability to produce models with limit cycles when modeling time series data that exhibit periodic behavior. In a physical context, limit cycles represent a stationary state of sustained oscillations, a satisfying behavior for any model of a time series with periodic behavior. Analysis of the Wolf sunspot numbers with MARS appears to give an improvement over existing nonlinear Threshold and Bilinear models.

20 Distribution/Availability of Abstract	21 Abstract Security Classification
[X] unclassified/unlimited [ ] same as report [ ] DTIC users	Unclassified

22a Name of Responsible Individual	22b Telephone (Include Area code)	22c Office Symbol
Lewis, P. A. W.	(408) 646-2283	OR/Lw

Nonlinear Modeling of Time Series using Multivariate Adaptive Regression Splines (MARS)

P.A.W. Lewis J.G. Stevens
Naval Postgraduate School, Monterey CA

SUMMARY

MARS is a new methodology, due to Friedman, for nonlinear regression modeling. MARS can be conceptualized as a generalization of recursive partitioning that uses spline fitting in lieu of other simple functions. Given a set of predictor variables, MARS fits a model in the form of an expansion in product spline basis functions of predictors chosen during a forward and backward recursive partitioning strategy. MARS produces continuous models for high dimensional data that can have multiple partitions and predictor variable interactions. Predictor variable contributions and interactions in a MARS model may be analyzed using an ANOVA style decomposition.

By letting the predictor variables in MARS be lagged values of a time series, one obtains a new method for nonlinear autoregressive threshold modeling of time series. A significant feature of this extension of MARS is its ability to produce models with limit cycles when modeling time series data that exhibit periodic behavior. In a physical context, limit cycles represent a stationary state of sustained oscillations, a satisfying behavior for any model of a time series with periodic behavior. Analysis of the Wolf sunspot numbers with MARS appears to give an improvement over existing nonlinear Threshold and Bilinear models.

Keywords: MULTIVARIATE ADAPTIVE REGRESSION SPLINES; NONLINEAR TIME SERIES MODELS; RECURSIVE PARTITIONING; REGRESSION SPLINES; THRESHOLD MODELS; ASTAR MODELS; WOLF SUNSPOT NUMBERS; LIMIT CYCLES

1 INTRODUCTION

Regression modeling is a frequently applied statistical technique that serves as a basis for studying and characterizing a system of interest. We use regression modeling to formulate a reasonable mathematical model of the relationship between the predictor and response variables of the system. The choice of a modeling form may be based on previous knowledge of the system or on considerations such as smoothness and continuity of the response and predictor variables.

Let y represent a single response variable that depends on a vector of p predictor variables $\boldsymbol{x}$ where $\boldsymbol{x} = (x_1, \ldots, x_v, \ldots, x_p)$. Assume we are given N samples of y and $\boldsymbol{x}$, namely $\{y_i, \boldsymbol{x}_i\}_{i=1}^N$ and that y is described by the regression model,

$$y = f(x_1, \ldots, x_p) + \epsilon \tag{1}$$

over some domain $D \subset \Re^p$, which contains the data. The function $f(\boldsymbol{x})$ reflects the true but unknown relationship between y and $\boldsymbol{x}$. The random additive error variable ϵ, which is assumed to have mean zero and variance σ_ϵ^2, reflects the dependence of y on quantities other than $\boldsymbol{x}$. The goal is to formulate a function $\hat{f}(\boldsymbol{x})$ that is a reasonable approximation of $f(\boldsymbol{x})$ over the domain D. If the correct parametric form of $f(\boldsymbol{x})$ is known, then we can use parametric regression modeling to estimate a finite number of unknown coefficients. However, in this paper the approach is nonparametric regression modeling (Eubank, 1988). We only assume that $f(\boldsymbol{x})$ belongs to a general collection of functions and rely on the data to determine the final model form and its associated coefficients.

In the first part of this paper we explain Multivariate Adaptive Regression Splines (MARS) (Friedman, 1988), a new method of flexible nonparametric regression modeling that appears to be an improvement over existing methodology when using moderate sample sizes N with dimension $p > 2$. Next, we introduce the use of MARS for modeling in a univariate time series context, x_τ for $\tau = 1, 2, \ldots N$, i.e., the predictor variables are the lagged values of the response variable x_τ. The result is a multivariate adaptive autoregressive spline model for the time series. Note that the discussion of MARS in this paper is a simple introduction that is only complete enough to motivate the extension to time series modeling with MARS. For further details on MARS see Friedman (1988).

In the regression context, MARS can be conceptualized as a generalization of a recursive partitioning strategy (Morgan and Sonquist, 1963; Breiman et al., 1984) which uses spline fitting in lieu of other simple functions. Given a set of predictor variables, MARS fits a model in the form of an expansion in product spline basis functions of predictors chosen during a forward and backward recursive partitioning strategy. Although MARS is a computationally intensive regression methodology, it can produce continuous models for high dimensional data that can have multiple partitions and predictor variable interactions. Predictor variable contributions and interactions in a MARS model may be analyzed using an ANOVA style decomposition.

Although MARS is capable of regression modeling in low dimensional environments $p \leq 2$, its primary advantages exist in higher dimensions. A difficulty with applying existing multivariate regression modeling methodologies to problems of dimension greater than two has been called the *curse-of-dimensionality* (Bellman, 1961). The *curse-of-dimensionality* describes the need for an exponential increase in sample size N for a linear increase in p, in order to densely populate higher dimensional spaces. MARS attempts to overcome the *curse-of-dimensionality* by exploiting the localized low-dimensional structure of the data used in constructing $\hat{f}(\boldsymbol{x})$.

With MARS, by letting the predictor variables be lagged values of a time series, one obtains a new method for nonlinear threshold modeling of time series we call ASTAR (Adaptive Spline Threshold Autoregression). We illustrate this methodology by applying ASTAR to simple autoregressive and nonlinear threshold models.

A significant feature of ASTAR is its ability to produce models with limit cycles when modeling time series data that exhibit periodic behavior. In a physical context, limit cycles represent a stationary state of sustained oscillations, a satisfying behavior for any model of a time series with periodic behavior. Our analysis of the Wolf sunspot numbers with ASTAR appears to improve existing nonlinear Threshold and Bilinear models.

In this paper the approach taken to explain MARS is geometric in nature; we focus on the iterative formation of overlapping subregions in the domain D of the predictor variables. Each subregion of the domain is associated with a product spline basis function. MARS approximates the unknown function $f(\boldsymbol{x})$ using the set of product spline basis functions associated with the overlapping subregions of the domain. To motivate the development of the MARS procedure, the next two sections briefly review recursive partitioning and regression splines. Section 4 is a discussion of Friedman's innovations used to develop MARS. An algorithm for implementing MARS is addressed in section 5. The application of MARS to the modeling of time series is discussed in Section 6.

2 RECURSIVE PARTITIONING (RP)

The origin of recursive partitioning regression methodology appears to date to the development and use of the AID (Automatic Interaction Detection) program by Morgan and Sonquist in the early 1960's. More recent extensions and contributions were made by Breiman et al. (1984). We explain recursive partitioning using recursive splitting of established subregions which is recast as an expansion in a set of basis functions. The latter explanation of recursive partitioning may be considered a precursor to MARS.

2.1 RP: Recursive Splitting of Established Subregions

Let the response variable y depend in some unknown way on a vector of p predictor variables $\boldsymbol{x} = (x_1, \ldots, x_p)$, that we model with (1). Assume we have N samples of y and $\boldsymbol{x}$, namely $\{y_i, \boldsymbol{x}_i\}_{i=1}^{N}$. Let $\{R_j\}_{j=1}^{S}$ be a set of S disjoint subregions of D such that $D = \bigcup_{j=1}^{S} R_j$. Given the subregions $\{R_j\}_{j=1}^{S}$, recursive partitioning estimates the unknown function $f(\boldsymbol{x})$ at $\boldsymbol{x}$ with

$$\hat{f}(\boldsymbol{x}) = \left\{ \hat{f}_j(\boldsymbol{x}) : \boldsymbol{x} \in R_j \right\}, \tag{2}$$

where the function $\hat{f}_j(\boldsymbol{x})$ estimates the true but unknown function $f(\boldsymbol{x})$ over the R_jth subregion of D. In recursive partitioning, $\hat{f}_j(\boldsymbol{x})$ is usually taken to be a constant (Morgan and Sonquist, 1963 and Breiman et al., 1984) although linear functions have been proposed without much success (Breiman and Meisel, 1976). For the purpose of explaining MARS $\hat{f}_j(\boldsymbol{x})$ is a constant function,

$$\hat{f}_j(\boldsymbol{x}) \;=\; c_j \;\; \forall \;\; \boldsymbol{x} \in R_j, \tag{3}$$

where each c_j is chosen to minimize the jth component of the residual-squared-error (badness-of-fit),

$$BOF[\hat{f}_j(\boldsymbol{x})] = \min_{c_j} \sum_{\boldsymbol{x}_i \in R_j} (y_i - c_j)^2. \tag{4}$$

Since the subregions of the domain D are disjoint, each c_j will be the sample mean of the y_i's whose $\{\boldsymbol{x}_i\}_{i=1}^N \in R_j$.

In general, the recursive partitioning model is the result of a 2-step procedure that starts with the single subregion $R_1 = D$. The first, or forward, step uses recursive splitting of established subregions to iteratively produce a large number of disjoint subregions $\{R_j\}_{j=2}^M$, for $M \geq S$, where M is chosen by the user. The second, or backward, step reverses the first step and trims the excess $(M - S)$ subregions using a criterion that evaluates both the model fit and the number of subregions in the model. The goal of the 2-step procedure is to use the data to select a good set of subregions $\{R_j\}_{j=1}^S$ together with the constant functions c_j that estimate $f(\boldsymbol{x})$ over each subregion of the domain.

To facilitate understanding of the recursive partitioning algorithm we examine the forward step procedure for an example problem using $p = 3$ predictor variables, and $M = 5$, the maximum number of forward step subregions. Let $v = 1, \ldots, p$ index the predictor variables and $k = 1, \ldots, n$ index the ordered sample values of a predictor variable x_v in subregion R_j. For our purposes we use $BOF_m = \sum_{j=1}^m BOF[\hat{f}_j(\boldsymbol{x})]$ as the forward step measure of fit for a recursive partitioning model with m subregions and restrict the set of candidate partition points to the actual sample values, $x_{v,k}$. Note that $x_{v,k}$ represents the kth ordered sample value of the vth predictor variable while x_v alone denotes the running values of the rth predictor variable. At the start of the forward step recursive partitioning algorithm, R_1 is the entire domain D and the single subregion estimate for $f(\boldsymbol{x})$ is

$$\hat{f}(\boldsymbol{x}) = \hat{f}_1(\boldsymbol{x}) = c_1 = \frac{1}{N} \sum_{i=1}^N y_i. \tag{5}$$

The forward step measure of fit for the single subregion recursive partitioning model is

$$BOF_1 = \sum_{i=1}^N (y_i - c_1)^2. \tag{6}$$

The initial recursion, $m = 2$, for the forward step algorithm selects a partition point t^* that best splits subregion R_1 into two disjoint sibling subregions. The method for discovering t^* is straightforward exhaustive search; evaluate *every* sample value $x_{v,k}$ (for $v = 1, \ldots, p; k = 1, \ldots, n$) as a candidate partition point to determine which one minimizes the remaining badness-of-fit for a $m = 2$ subregion model. For example, let $t = x_{1,15}$ identify a candidate partition point for predictor variable x_1. The area in parent subregion R_1 to the left of t, $x_1 < t$, resides in proposed sibling subregion $R_{1,l}$. The area to the right of t, $t \leq x_1$, resides in proposed

sibling subregion $R_{1,r}$. Given the proposed split of R_1 along $t = x_{1,15}$, we evaluate the model using BOF_m for a $m = 2$ subregion model, i.e.,

$$BOF_2 = \min_{c_l} \sum_{\boldsymbol{x}_i \in R_{1,l}} (y_i - c_l)^2 \; + \; \min_{c_r} \sum_{\boldsymbol{x}_i \in R_{1,r}} (y_i - c_r)^2. \tag{7}$$

Using the indices v and k the exhaustive search sequentially evaluates all possible partition points for each predictor variable in R_1, which here is equal to D.

For our example problem, let the partition point $t^* = x_{2,25}$ identify the split of subregion R_1 that minimizes the forward step fit criterion BOF_m for a $m = 2$ subregion recursive partitioning model. We use $x_{2,25}$ to create two new disjoint subregions during the split and elimination of the old parent region R_{1*}. First, the area in parent subregion R_{1*} to the left of t^* i.e., $x_2 < t^*$ is assigned to sibling subregion R_2 while the area to the right of t^* i.e., $t^* \leq x_2$ is reconstituted as subregion R_1. The creation of the two new disjoint subregions R_1 and R_2 and the elimination of the old parent subregion R_{1*} increase by one the number of disjoint subregions that partition D and finish the initial recursion of the forward step procedure. Thus, the two subregion recursive partitioning estimate of $f(\boldsymbol{x})$ for our example problem is

$$\hat{f}(\boldsymbol{x}) = \{c_j : \boldsymbol{x} \in R_j \text{ for } j = 1, 2\}, \tag{8}$$

where, since we are splitting the domain D on only one dimension, namely x_2,

$$\boldsymbol{x} \in \begin{cases} R_1 & \text{if} \quad x_2 \geq x_{2,25} \\ R_2 & \text{if} \quad x_2 < x_{2,25}. \end{cases}$$

Note that the form of the recursive partitioning model (2) did not change during the recursion, only the number of disjoint subregions that partition D.

The recursions $m = 3, \ldots, M = 5$ of the forward step algorithm, are a repeat of the first recursion with one exception. The exhaustive search is now conducted to identify the best split for *one and only one* of the subregions from the current $m-1$ subregion model. Each recursion's partition point t^* is selected as before, after an evaluation of all potential partition points for each predictor variable in the existing subregions $\{R_j\}_{j=1}^{m-1}$ of the model. The recursive splitting continues until the domain D is partitioned into $M = 5$ disjoint subregions $\{R_j\}_{j=1}^{5}$. Upon completion of the forward step recursive partitioning algorithm, a backward step algorithm trims excess subregions using a criterion that evaluates both fit and the number of subregions in the model. See (Friedman, 1988) for a discussion of the backward step algorithm. Completion of the backward step procedure results in the final recursive partitioning model with $\{R_j\}_{j=1}^{S}$ subregions.

2.2 RP: An Expansion in a Set of Basis Functions

While the intuitive approach to understanding recursive partitioning is through recursive splitting, it is recast now in a form that provides a reference for explaining the MARS methodology. The central idea is to formulate the recursive partitioning model as an additive model of functions from disjoint subregions. Also, we associate the operation of subregion splitting with the operation of step function multiplying. The new approach approximates the unknown function $f(\boldsymbol{x})$ at $\boldsymbol{x}$ with an expansion in a set of basis functions from disjoint subregions $\{R_j\}_{j=1}^{S}$,

$$\hat{f}(\boldsymbol{x}) = \sum_{j=1}^{S} c_j B_j(\boldsymbol{x}), \tag{9}$$

where

$$B_j(\boldsymbol{x}) = I[\boldsymbol{x} \in R_j],$$

and $I[x]$ is an indicator function with value 1 if its argument is true and 0 otherwise. The constant function c_j estimates the true but unknown function $f(\boldsymbol{x})$ over the R_jth subregion of D and $B_j(\boldsymbol{x})$ is a basis function that indicates membership in the R_jth subregion of D. We call $B_j(\boldsymbol{x})$ a basis function because it restricts contributions for $\hat{f}(\boldsymbol{x})$ to those values of $\boldsymbol{x}$ in the R_jth subregion of D. The approximation of the unknown function $f(\boldsymbol{x})$ at $\boldsymbol{x}$ in (2) and (9) are equivalent; the subregions $\{R_j\}_{j=1}^{S}$ are the same disjoint subregions of the domain D and the constant functions $\{c_j\}_{j=1}^{S}$ are the same constant functions that estimate $f(\boldsymbol{x})$ over each subregion.

During each search for a partition of a subregion R_j using an expansion in a set of basis functions (9), the selection of a candidate partition point creates a particular functional form for $\hat{f}(\boldsymbol{x})$ that we call g in the following algorithm. Let $H[\eta]$ be a step function that returns a value of 1 if η is positive and 0 otherwise. Following Friedman (1988), an algorithm to implement the forward step recursive partitioning procedure using an expansion in a set of basis functions is:

Recursive Partitioning Algorithm (Forward Step) (10)

$R_1 = D$, $B_1(\boldsymbol{x}) = 1$ (a)
For each subregion R_m, $m = 2$ to M do: (b)
  bof$^* = \infty$, $j^* = 0$, $v^* = 0$, $t^* = 0$ (c)
  For each established subregion R_j, $j = 1$ to $m-1$ do: (d)
    For each predictor variable x_v in R_j, $v = 1$ to p do: (e)
      For each data value $x_{v,k}$ in R_j, $t = x_{v,k=1}$ to $x_{v,k=n}$ do: (f)
        $g = \left(\sum_{d \neq j} c_d B_d(\boldsymbol{x})\right) + c_m B_j(\boldsymbol{x}) H[t - x_v] + c_j B_j(\boldsymbol{x}) H[x_v - t]$ (g)
        bof = BOF_m (h)
        if bof < bof* then bof* = bof ; $j^* = j$; $v^* = v$; $t^* = t$ end if (i)
      end for
    end for
  end for
  $R_m \leftarrow R_{j^*} H[t^* - x_{v^*}]$ (j)
  $R_{j^*} \leftarrow R_{j^*} H[x_{v^*} - t^*]$ (k)
end for
end

The forward step recursive partitioning algorithm is initialized with the first subregion R_1 equal to the entire domain D (10a). The outer loop (10b) controls the iterative creation of the subregions $\{R_m\}_{m=2}^{M}$. Next, the dummy variables (10c) for the evaluation of the fit procedure bof*, region j^*, predictor variable v^*, and partition point t^* are initialized in preparation for identifying the next partition of an established subregion $\{R_j\}_{j=1}^{m-1}$. The three nested inner loops (10d-10f) perform the exhaustive search for the next partition point by iteratively searching across all established subregions (10d), all predictor variables (10e), and all values of the predictor variables in the jth subregion (10f). Given the investigation of a partition point t for a predictor variable x_v in subregion R_j, the function g (10g), with parameter vector $\boldsymbol{c} = (c_1, \ldots, c_m)$, is the current candidate for a recursive partitioning model estimate of $f(\boldsymbol{x})$ in the mth iteration of the forward step procedure. The first term in (10g) includes all subregions except subregion R_j. The last two terms in (10g),

$$c_m B_j(\boldsymbol{x}) H[t - x_v] + c_j B_j(\boldsymbol{x}) H[x_v - t],$$

reflect the proposal to divide the parent subregion R_j into two disjoint sibling subregions using the step functions $H[t - x_v]$ and $H[x_v - t]$ to identify each $\boldsymbol{x}$'s location with respect to the partition point t. Next, BOF_m (10h) is the forward step measure of fit that evaluates the function g with respect to the data. Information for the best yet discovered partition, predictor variable, and subregion (10i) is retained as the search continues for the best partition of an established subregion $\{R_j\}_{j=1}^{m-1}$ in the mth iteration. Completion of the mth iteration's search results in the division (and elimination) of the old parent subregion R_{j^*} into two disjoint sibling subregions (10j and 10k) based on x_{v^*}'s location with respect to the partition point t^*. The iterations continue until the domain D is partitioned into M disjoint subregions $\{R_j\}_{j=1}^{M}$.

Each basis function $B_j(\boldsymbol{x})$ identifies membership in the R_jth subregion of D and is the result of the product of step functions whose partition points define the subregion R_j. For example, let R_5 be a subregion from the sequence of step functions $H[\boldsymbol{x}_1 - t_1^*], H[t_2^* - \boldsymbol{x}_2], H[\boldsymbol{x}_2 - t_3^*]$ and $H[t_4^* - \boldsymbol{x}_1]$ where $\{t_i^*\}_{i=1}^4$ is 0,1,0,1 respectively. Then the basis function $B_5(\boldsymbol{x})$ is,

$$B_5(\boldsymbol{x}) = H[\boldsymbol{x}_1 - 0] \times H[1 - \boldsymbol{x}_2] \times H[\boldsymbol{x}_2 - 0] \times H[1 - \boldsymbol{x}_1], \tag{11}$$

which defines the subregion R_5 as a unit square in $\Re^2$. The basis function $B_5(\boldsymbol{x}) = 1$ if $0 \leq \boldsymbol{x}_1 < 1$ and $0 \leq \boldsymbol{x}_2 < 1$ and is 0 otherwise.

In recursive partitioning the subregions $\{R_j\}_{j=1}^S$ *are disjoint.* Each data point $\boldsymbol{x}$ is only a member of one subregion R_j. Therefore, the estimate of $f(\boldsymbol{x})$ over subregion R_j is restricted to the functional form for $\hat{f}_j(\boldsymbol{x})$. However, as we will address in section 4, MARS *has overlapping subregions.* The estimate of $f(\boldsymbol{x})$ over subregion R_j may be obtained as a sum of multiple functional forms.

Recursive partitioning is a very powerful methodology that is rapidly computed, especially if $\hat{f}_j(\boldsymbol{x})$ is the constant c_j. Each forward step of the algorithm (10) partitions *one and only one* subregion of the domain on an influential variable $\boldsymbol{x}_{v^*}$. This procedure increasingly localizes the activity of the predictor variables with respect to the response variable y. However, in general, there are several drawbacks to using recursive partitioning as a regression modeling technique.

- Recursive partitioning models have disjoint subregions and are usually discontinuous at subregion boundaries. This is disconcerting if we believe $f(\boldsymbol{x})$ is continuous.

- Recursive partitioning has an innate inability to adequately estimate linear or additive functions. This is due to the recursive division of established subregions during the forward step procedure that automatically produces predictor variable interactions unless all successive partitions occur on the same predictor variable.

- The form of the recursive partitioning model (9), an additive combination of functions of predictor variables in disjoint regions, makes estimation of the true form of the unknown function $f(\boldsymbol{x})$ difficult for large p.

3 REGRESSION SPLINES

The development of a regression spline model offers another method for explaining MARS. Silverman (1985) views spline functions as an attractive approach to modeling that may be thought of as a span between parametric and nonparametric regression methodology. For simplicity define a qth order polynomial function of $\boldsymbol{x} \in D \subset \Re^1$

with coefficients c_l as

$$p_q(x) = \sum_{l=0}^{q} c_l x^l \quad \text{for} \quad x \in D. \tag{12}$$

Polynomials such as (12) are smooth and easy to manipulate. However, fitting data with a polynomial model may require higher order terms that may have unacceptable fluctuations. This leads us to divide the domain D into smaller subregions R_j to permit the use of polynomial functions of relatively low order.

Let $[a,b] = D \subset \Re^1$ and $\Delta_S = \{t_1, \ldots, t_{S-1}\}$ denote an ordered partition of $[a,b]$ into S disjoint subregions i.e., $a = t_0 < t_1 < \ldots < t_{S-1} < t_S = b$. Denote the S disjoint subregions as $R_j = [t_{j-1}, t_j]$, for $j = 1, \ldots, S$. Let $C^q[D]$ represent the set of all continuous functions in D whose $q-1$ derivatives are also continuous. Using j to index the subregions we define a spline function as a set of S piecewise qth order polynomial functions whose function values and first $q-1$ derivatives agree at their partition points i.e.,

$$\begin{aligned} s^q_{\Delta_S}(x) &= \sum_{j=1}^{S} p_{q,j}(x)\, I[x \in R_j] \\ \text{with the restriction that} \quad s^q_{\Delta_S}(x) &\in C^q[D]. \end{aligned} \tag{13}$$

There are several approaches for implementing splines within a regression setting (Wegman and Wright, 1983). One approach is the piecewise regression spline model,

$$y = s^q_{\Delta_S}(x) + \epsilon, \tag{14}$$

where again ϵ is assumed to have mean zero and variance σ^2_ϵ, and $s^q_{\Delta_S}(x)$ from (13) estimates $f(x)$.

Given a set of partitions points Δ_S, Smith (1979) has shown that a different and more useful regression spline model may be written using plus (+) functions. The plus function is defined as

$$u_+ = \begin{cases} u & \text{if} \quad u > 0 \\ 0 & \text{if} \quad u \leq 0. \end{cases} \tag{15}$$

Again, let $[a,b] = D \subset \Re^1$. However, we now let $\Delta_{So} = \{t_1, \ldots, t_{S-1}\}$ define an ordered partition of $[a,b]$ into S *overlapping subregions* and denote the S *overlapping subregions* as $R_j = [t_{j-1}, t_S]$, for $j = 1, \ldots, S$. Let l index the order of the polynomial terms in each subregion of the domain and c_{jl} denote the coefficients for the lth term of the polynomial function in the $(j+1)$st subregion of a spline model. Using plus functions results in a truncated regression spline model functionally equivalent to the piecewise regression spline model (13),

$$y = \sum_{l=0}^{q} c_{0l} x^l + \sum_{j=1}^{S-1} c_{jq}[(x - t_j)_+]^q + \epsilon, \tag{16}$$

where ϵ is assumed to have mean zero and variance σ^2_ϵ. Since the partitions points of the set Δ_{So} are ordered, the number of overlapping truncated spline functions with nonzero values increases by 1 as we move to the right

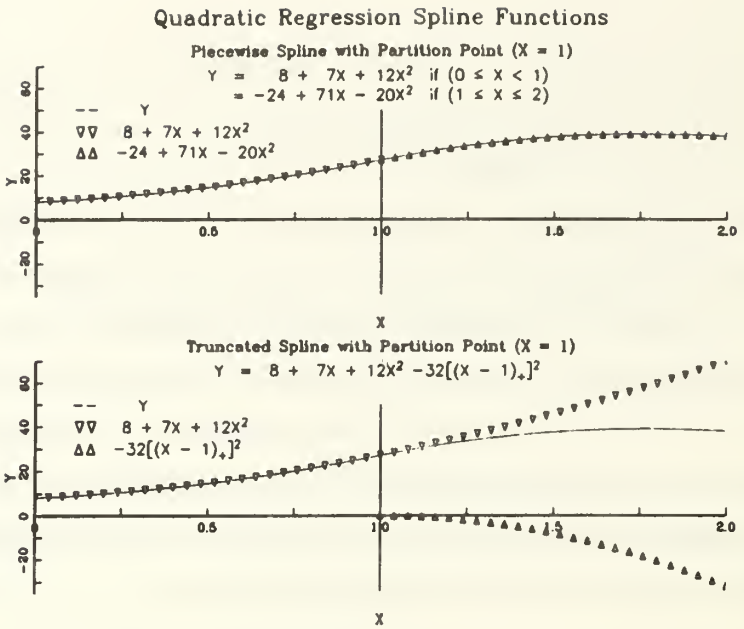

Figure 1: The different forms for a piecewise (13) and truncated (16) spline function using $q = 2$ order splines over the region $D = [0, 2]$ with a single partition point, at $X = 1$.

and cross each partition point t_j. Figure 1 compares the different forms for a $q = 2$ order piecewise (13) and truncated (16) spline function.

The key point of this section is that once the number and the values of the partition points $\{t_j\}_{j=1}^{S-1}$ are fixed, the qth order truncated regression spline model (16) with those partition points is a linear model whose coefficients $\boldsymbol{c}$ may be determined by straightforward least squares regression. However, the major difficulty in implementing a qth order regression spline model is choosing the number and values of the partition points.

We have defined regression spline models in $\Re^1$. The extension to higher dimensions for $p > 1$ predictor variables is usually accomplished using products of univariate spline functions. However, products of univariate spline functions suffer from the *curse-of-dimensionality* discussed previously. From the perspective of regression splines, MARS attempts to overcome the *curse-of-dimensionality* by using a modified recursive partitioning strategy to select partitions of the domain. This permits MARS to exploit the localized, low-dimensional structure of the data using $q = 1$ truncated, multidimensional regression spline functions.

4 FRIEDMAN'S INNOVATIONS FOR RECURSIVE PARTITIONING

Recursive partitioning and regression splines have tremendous power for modeling in high dimensional environments. Each approach also presents difficulties when applied; recursive partitioning has discontinuities,

variable interactions and poor model interpretation, and regression splines battle the *curse-of-dimensionality* and lack a methodology to optimally select its many parameters.

Two aspects of the recursive partitioning algorithm (10) contribute to the difficulties of its application in a high dimensional setting. The iterative division and elimination of the parent region when creating its sibling subregions causes difficulty in estimating linear and additive functions. The discontinuous nature of the step function $H[\eta]$ when applied in each linear regression of the forward step recursive partitioning algorithm (10g) causes the lack of continuity. Together, these characteristics make interpretation of the recursive partitioning model difficult at best.

To overcome recursive partitioning's difficulty in estimating linear and additive functions, Friedman proposes that the parent region is not eliminated (as in recursive partitioning) during the creation of its sibling subregions. Thus, in future iterations both the parent and its sibling subregions are eligible for further partitioning. An immediate result of retaining parent regions is overlapping subregions of the domain. Also, each parent region may have multiple sets of sibling subregions. With this modification, recursive partitioning can produce linear models with the repetitive partitioning of the initial region R_1 by different predictor variables. Additive models with functions of more than one predictor variable can result from successive partitioning using different predictor variables. This modification also allows for multiple partitions of the same predictor variable from the same parent region.

Maintaining the parent region in a modified recursive partitioning algorithm results in a class of models with greater flexibility than permitted in recursive partitioning. However, the modified approach is still burdened with the discontinuities caused by the step function $H[\eta]$. To alleviate this difficulty, Friedman proposes to replace the step function $H[\eta]$ in the model formulation step (10g) with $q = 1$ order (i.e. linear) regression splines in the form of left (-) and right (+) truncated splines. Let $\boldsymbol{r}_m$ represent a 2-tuple associated with the R_mth subregion whose components identify the direction (left or right), specific predictor variable, and partition point used to create subregion R_m from its parent region. A left and right truncated spline for creating the R_mth and R_{m+1}st subregion from the parent region R_j with a partition point at $x_v = t$ is defined as

$$T_{j,\boldsymbol{r}_m}(\boldsymbol{x}) = [(t - x_v)_+]^{q=1} = (t - x_v)_+ \qquad \text{and} \qquad T_{j,\boldsymbol{r}_{m+1}}(\boldsymbol{x}) = [(x_v - t)_+]^{q=1} = (x_v - t)_+, \tag{17}$$

where $\boldsymbol{r}_m = (-v, t)$ and $\boldsymbol{r}_{m+1} = (+v, t)$ and $m > j$. The additional subscripts j and m, or j and $m + 1$, provide a necessary audit trail for products of truncated splines when interactions are allowed among multiple predictor variables. *Note that the truncated spline functions act in only one dimension although their argument is a vector of predictor variables.*

A modeling approach using linear truncated splines (17) creates a continuous approximating function $\hat{f}(\boldsymbol{x})$ with discontinuities in the first partial derivative of $\hat{f}(\boldsymbol{x})$ at the partition points of each predictor variable in the model. The argument for using *linear* truncated splines (17) is that there is little to be gained in flexibility, and much to lose in computational speed by imposing continuity beyond the function $\hat{f}(\boldsymbol{x})$. Linear truncated splines allow rapid updating of the regression model and its coefficients during each exhaustive search for the next partition of an established subregion. The placement of additional partitions may be used to compensate for the loss of flexibility in using linear truncated splines to estimate $f(\boldsymbol{x})$ over a subregion of the domain.

Implementation of the modifications proposed above to the recursive partitioning algorithm avoids its identified difficulties and results in the MARS algorithm. The MARS algorithm produces a linear ($q = 1$) truncated spline model (16) with overlapping subregions $\{R_j\}_{j=1}^{S}$ of the domain D. Each overlapping subregion of a MARS model is defined by the partition points of the predictor variables from an ordered sequence of linear truncated splines.

Define the product basis function $K_m(\boldsymbol{x})$ as the ordered sequence of truncated splines associated with subregion R_m. The first term of every product basis function is $T_{0,\boldsymbol{r}_1}(\boldsymbol{x}) = 1$, the initialization function associated with R_1. Each additional truncated spline represents the iterative partitioning of a parent region into a sibling subregion. For example, assume the sequence of ordered truncated splines for the parent region R_7 is (1,3,7), which is split using $T_{\boldsymbol{r}_7,\boldsymbol{r}_m}(\boldsymbol{x})$ to create subregion R_m. The product basis function $K_m(\boldsymbol{x})$ associated with the R_mth subregion for this example is

$$K_m(\boldsymbol{x}) = \underbrace{T_{0,\boldsymbol{r}_1}(\boldsymbol{x}) \times T_{\boldsymbol{r}_1,\boldsymbol{r}_3}(\boldsymbol{x}) \times T_{\boldsymbol{r}_3,\boldsymbol{r}_7}(\boldsymbol{x})}_{K_7(\boldsymbol{x})} \times T_{\boldsymbol{r}_7,\boldsymbol{r}_m}(\boldsymbol{x}). \tag{18}$$

where $m > 7$.

To evaluate $K_m(\boldsymbol{x})$ at $\boldsymbol{x}$ requires the evaluation of each truncated spline in the product basis function at $\boldsymbol{x}$. If any of the truncated spline evaluations at $\boldsymbol{x}$ are zero, then $K_m(\boldsymbol{x})$ at $\boldsymbol{x}$ is 0. Otherwise, the evaluation of $K_m(\boldsymbol{x})$ at $\boldsymbol{x}$ is the product of the truncated splines at $\boldsymbol{x}$. For example, let the ordered truncated splines for $R_5 \in \Re^3$ be (1,2 and 5) with $\boldsymbol{r}_2 = (2, 3)$ and $\boldsymbol{r}_5 = (-3, 1)$. The product basis function associated with R_5 is

$$\begin{aligned} K_5(\boldsymbol{x}) = &\ T_{0,\boldsymbol{r}_1}(\boldsymbol{x}) \times T_{\boldsymbol{r}_1,\boldsymbol{r}_2}(\boldsymbol{x}) \times T_{\boldsymbol{r}_2,\boldsymbol{r}_5}(\boldsymbol{x}) \\ = &\quad 1 \times (x_2 - 3)_+ \times (1 - x_3)_+ \quad = \begin{cases} (x_2 - 2)(1 - x_3) & \text{if } x_2 > 2 \text{ and } x_3 < 1 \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

If $\boldsymbol{x}_1 = \{5, 4, 0\} \in R_5$ and $\boldsymbol{x}_2 = \{4, 3, 6\} \ni R_5$, then $K_5(\boldsymbol{x}_1) = 2$ and $K_5(\boldsymbol{x}_2) = 0$.

The level of interaction of the predictor variables associated with R_j is the number of truncated splines (without $T_{0,\boldsymbol{r}_1}(\boldsymbol{x})$) in a product basis function $K_j(\boldsymbol{x})$. A one term product basis function represents a truncated linear relationship of its predictor variable while a two term product basis function represents a truncated 2-way

interaction and so on. The number and level of interactions in a MARS model are only limited by the data and the level of interactions permitted by the MARS algorithm.

The MARS estimate of the unknown function $f(\boldsymbol{x})$ is

$$\hat{f}(\boldsymbol{x}) = \sum_{j=1}^{S} c_j K_j(\boldsymbol{x}), \tag{19}$$

where $\hat{f}(\boldsymbol{x})$ is an additive function of the product basis functions $\{K_j(\boldsymbol{x})\}_{j=1}^{S}$ associated with the subregions $\{R_j\}_{j=1}^{S}$. As in recursive partitioning the objective of the *forward step* MARS algorithm is to iteratively adjust the vector of coefficient values to best fit the data while identifying the subregions $\{R_j\}_{j=1}^{M}$, for $M \geq S$, whose product basis functions approximate $f(\boldsymbol{x})$ based on data at hand. And again, as in the recursive partitioning procedure, it makes sense to follow the forward step procedure with a backward step trimming procedure to remove the excess $(M - S)$ subregions whose product basis functions no longer sufficiently contribute to the accuracy of the fit.

5 FORWARD STEP MARS ALGORITHM

The MARS forward step algorithm results from applying the modifications addressed in Section 4 to the forward step recursive partitioning algorithm (10). Again we initialize $R_1 = D$. However, in MARS we create two new subregions R_m and R_{m+1} and maintain the parent region R_{j^*} during each partition. Also, MARS restricts each sequence of truncated splines from having more than one partition per predictor variable because this creates a nonlinear spline function i.e., one with $q > 1$. MARS enforces this restriction, during the search for the next best partition of a subregion R_j, by excluding from consideration for a partition point any predictor variable already included in the product basis function $K_j(\boldsymbol{x})$. The most notable difference between the RP and MARS algorithms occurs in forming the MARS model. Again following Friedman (1988), the product basis functions $\{K_j(\boldsymbol{x})\}_{j=1}^{m}$ given at (18) and the truncated splines $T_{\boldsymbol{r}_j,\boldsymbol{r}_m}(\boldsymbol{x})$ and $T_{\boldsymbol{r}_j,\boldsymbol{r}_{m+1}}(\boldsymbol{x})$ given at (17) replace the basis functions $\{B_j(\boldsymbol{x})\}_{j=1}^{m}$ and the step functions $H[t - x_v]$ and $H[x_v - t]$ in the forward step recursive partitioning algorithm (10g) respectively.

MARS Forward Step Algorithm (20)

$R_1 = D,\ T_{0,\boldsymbol{r}_1}(\boldsymbol{x}) = 1$ (a)
For each subregion R_m, $m = 2$ to M do: (b)
  bof$^* = \infty,\ j^* = 0,\ v^* = 0,\ t^* = 0$ (c)
  For each established subregion R_j, $j = 1$ to $m-1$ do: (d)
    For each predictor variable x_v in R_j, $v = 1$ to p such that $v \ni K_j(\boldsymbol{x})$ do: (e)
      For each data value $x_{v,k}$ in R_j, $t = x_{v,k=1}$ to $x_{v,k=n}$ do: (f)
        $g = (\sum_d c_d K_d(\boldsymbol{x})) + c_m K_j(\boldsymbol{x}) T_{\boldsymbol{r}_j,\boldsymbol{r}_m}(\boldsymbol{x}) + c_{m+1} K_j(\boldsymbol{x}) T_{\boldsymbol{r}_j,\boldsymbol{r}_{m+1}}(\boldsymbol{x})$ (g)
        bof= BOF_m (h)
        if bof $<$ bof* then bof* = bof ; $j^* = j$; $v^* = v$; $t^* = t$ end if (i)
      end for
    end for
  end for
  $R_m \leftarrow R_{j^*} H[(t^* - x_{v^*})]$ (j)
  $R_{m+1} \leftarrow R_{j^*} H[(x_{v^*} - t^*)]$ (k)
  $m \leftarrow m + 2$ (l)
end for
end

To characterize the MARS procedure we use the example discussed in section 2 with $p = 3$ predictor variables, and $M = 5$, the maximum number of forward step partitions. The MARS algorithm parallels the recursive partitioning algorithm except for the modifications discussed in section 4. At the start of the MARS forward step algorithm for our example problem, the initial subregion is again the entire domain i.e., $R_1 = D$. Thus, the single subregion MARS estimate of $f(\boldsymbol{x})$ is identical to the recursive partitioning estimate,

$$\hat{f}(\boldsymbol{x}) \;=\; c_1 K_1(\boldsymbol{x}) \;=\; c_1 T_{0,\boldsymbol{r}_1}(\boldsymbol{x}) \;=\; c_1 \;=\; \frac{1}{N}\sum_1^N y_i. \tag{21}$$

Again, let the exhaustive search in the first iteration of MARS identify the best partition of R_1 as $t^* = x_{2,25}$. Continuing, the three subregion MARS estimate of $f(\boldsymbol{x})$ obtained at the second step (first partition at $t^* = x_{2,25}$) is, with $T_{0,\boldsymbol{r}_1}(\boldsymbol{x}) = 1$,

$$\begin{aligned}\hat{f}(\boldsymbol{x}) \;&= c_1\, K_1(\boldsymbol{x}) + c_2\, K_2(\boldsymbol{x}) + c_3\, K_3(\boldsymbol{x}) \\ &= c_1\, T_{0,\boldsymbol{r}_1}(\boldsymbol{x}) + c_2\, T_{0,\boldsymbol{r}_1}(\boldsymbol{x})\, T_{\boldsymbol{r}_1,\boldsymbol{r}_2}(\boldsymbol{x}) + c_3\, T_{0,\boldsymbol{r}_1}(\boldsymbol{x})\, T_{\boldsymbol{r}_1,\boldsymbol{r}_3}(\boldsymbol{x}) \\ &= c_1 + c_2\,(t^* - x_2)_+ + c_3\,(x_2 - t^*)_+,\end{aligned} \tag{22}$$

$$\text{where } \boldsymbol{x} \in \begin{cases} R_1 & \text{if} \quad \boldsymbol{x} \in D \\ R_2 & \text{if} \quad x_2 < x_{2,25} \text{ and } \boldsymbol{x} \in R_1 \\ R_3 & \text{if} \quad x_2 \geq x_{2,25} \text{ and } \boldsymbol{x} \in R_1. \end{cases}$$

In the next iteration of the forward step MARS algorithm the best partition point will occur within the subregions R_1, R_2 or R_3 and as in recursive partitioning, with one exception, will be chosen after evaluation of

all potential partition points for each predictor variable within the three subregions. The exception, as discussed previously, prevents another partition on x_2 in R_2 or R_3 because it would create a nonlinear truncated spline function. With $M = 5$ the forward step of the MARS algorithm will be complete after a second partition in D. The final forward step MARS estimate of $f(\boldsymbol{x})$ for our example will include all terms in (22) and the additional two terms generated by the second partition. The model will have 5 single term product spline functions (excluding $T_{0,\boldsymbol{r}_1}(\boldsymbol{x})$) if the second partition occurs in R_1 while the model will have 3 single term product spline functions and two 2-way product spline functions if the second partition occurs in R_2 or R_3.

After the backward trimming procedure, the final MARS model retains the form of (19) with c_1 the coefficient of the product basis function $K_1(\boldsymbol{x})$ and the remaining terms the coefficients and product basis functions that survive the MARS backward step subregion deletion strategy. To provide an insight of predictor variable relationships we can rearrange the final MARS estimate of $f(\boldsymbol{x})$ in an ANOVA style decomposition,

$$\hat{f}(\boldsymbol{x}) = c_1 + \sum_{V=1} c_j K_j(\boldsymbol{x}) + \sum_{V=2} c_j K_j(\boldsymbol{x}) + \ldots \tag{23}$$

where V indexes the number of truncated splines (excluding $T_{0,\boldsymbol{r}_1}(\boldsymbol{x})$) in the product basis function $\{K_j(\boldsymbol{x})\}_{j=1}^{S}$. This method identifies any and all contributions to $\hat{f}(\boldsymbol{x})$ by variables of interest. Product basis functions with the index $V = 1$ reflect truncated linear trends and those with the index $V = 2$ reflect truncated 2-way interactions, etc. The ANOVA style decomposition (23) identifies which variables enter the model, whether they are purely additive, or are involved in interactions with other variables. Analysis of the ANOVA style decomposition facilitates interpretation of the MARS model.

MARS uses residual-squared-error in the forward and backward steps of the algorithm to evaluate model fit and compare partition points because of its attractive computational properties. The actual backward fit criterion is a modified form of generalized cross validation (GCV) first proposed by Craven and Wahba (1979). The GCV criterion of a MARS model with the subregions $\{R_j\}_{j=1}^{M}$ is,

$$GCV(M) = \frac{\frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{f}_M(\boldsymbol{x}_i)]^2}{[1 - \frac{C(M)}{N}]^2}, \tag{24}$$

where $C(M)$ is a complexity cost function, increasing in M, which accounts for the increasing model complexity due to the sequential partition of D into the subregions $\{R_j\}_{j=1}^{M}$. The numerator of the GCV criteria is the average residual-squared-error and the denominator is a penalty term that reflects model complexity.

6 NONLINEAR MODELING OF TIME SERIES USING MARS

Most research in and applications of time series modeling and analysis are concerned with linear models. However, nonlinear time dependent systems abound that are not adequately handled by linear models. For

these systems we need to consider general classes of nonlinear models that readily adapt to the precise form of a nonlinear system of interest (Priestley, 1988). By letting the predictor variables for the τth value in a time series $\{x_\tau\}$ be $x_{\tau-1}, x_{\tau-2}, \ldots, x_{\tau-p}$, and combining these predictor variables into a linear additive function, one gets the well known linear AR(p) time series models. What happens if we use the MARS methodology to model the effect on x_τ by $x_{\tau-1}, x_{\tau-2}, \ldots, x_{\tau-p}$? The answer is that we still obtain autoregressive models of the effect on x_τ by $x_{\tau-1}, x_{\tau-2}, \ldots, x_{\tau-p}$, however, these models can have nonlinear terms from lagged predictor variable thresholds and interactions. We now pursue the form and analysis of these nonlinear models.

Threshold models (models with partition points) are a class of nonlinear models that emerge naturally as a result of changing physical behavior. Within the domain of the predictor variables, different model forms are necessary to capture changes to the relationship between the predictor and response variables. Tong (1983) provides one threshold modeling methodology for this behavior (TAR – Threshold Autoregression) that identifies piecewise linear pieces of nonlinear functions over disjoint subregions of the domain D i.e., identify linear models within each disjoint subregion of the domain. One application of Tong's threshold modeling methodology is for nonlinear systems thought to possess periodic behavior in the form of stationary sustained oscillations (limit cycles). Tong's threshold methodology has tremendous power and flexibility for modeling of many times series. However, unless Tong's methodology is constrained to be continuous, it creates disjoint subregion models that are discontinuous at subregion boundaries.

With MARS, by letting the predictor variables be lagged values of a time series, one admits a more general class of continuous nonlinear threshold models than permitted by Tong's TAR approach. We call the methodology for developing this class of nonlinear threshold models ASTAR (Adaptive Spline Threshold Autoregression). The fact that one obtains a more general class of continuous nonlinear threshold models can be shown using a simple example. Let X_τ for $\tau = 1, \ldots, N$, be a time series we wish to model with ASTAR using, for example, $p = 3$ lagged predictor variables namely, $X_{\tau-1}, X_{\tau-2}$ and $X_{\tau-3}$. Each forward step of the ASTAR algorithm selects *one and only one* set of new terms for the ASTAR model from the candidates specified by previously selected terms of the model. The sets of candidates for the initial forward step of the ASTAR algorithm for our example problem is

$$\begin{aligned} &(X_{\tau-1} - t^*)_+ \text{ and } (t^* - X_{\tau-1})_+, \text{ or} \\ &(X_{\tau-2} - t^*)_+ \text{ and } (t^* - X_{\tau-2})_+, \text{ or} \\ &(X_{\tau-3} - t^*)_+ \text{ and } (t^* - X_{\tau-3})_+, \end{aligned} \tag{25}$$

for some partition point (threshold) t^* in the individual domain of the lagged predictor variables. For our

example problem, assume that ASTAR selects the lagged predictor variable $X_{\tau-2}$ with threshold value $t^* = t_1$ i.e., $(X_{\tau-2} - t_1)_+$ and $(t_1 - X_{\tau-2})_+$ are the initial terms (other than the constant) in the ASTAR model. The sets of candidates for the second forward step of the ASTAR algorithm includes all candidates in (25) and the new sets of candidates:

$$\begin{aligned}
&(X_{\tau-1} - t^*)_+(X_{\tau-2} - t_1)_+ \text{ and } (t^* - X_{\tau-1})_+(X_{\tau-2} - t_1)_+, \text{ or} \\
&(X_{\tau-3} - t^*)_+(X_{\tau-2} - t_1)_+ \text{ and } (t^* - X_{\tau-3})_+(X_{\tau-2} - t_1)_+, \text{ or} \\
&(X_{\tau-1} - t^*)_+(t_1 - X_{\tau-2})_+ \text{ and } (t^* - X_{\tau-1})_+(t_1 - X_{\tau-2})_+, \text{ or} \\
&(X_{\tau-3} - t^*)_+(t_1 - X_{\tau-2})_+ \text{ and } (t^* - X_{\tau-3})_+(t_1 - X_{\tau-2})_+,
\end{aligned} \tag{26}$$

due to the initial selection of $(X_{\tau-2} - t_1)_+$ and $(t_1 - X_{\tau-2})_+$. The sets of candidates for each subsequent forward step of the ASTAR algorithm is nondecreasing in size and is based on previously selected terms of the model. As discussed in Section 4, the forward step algorithm is followed by a backward step algorithm that trims the excess $(M - S)$ terms from the model.

By modeling univariate time series using ASTAR we overcome the limitations of Tong's approach. The ASTAR methodology creates threshold models that are naturally continuous in the domain of the predictor variables, allow interactions among lagged predictor variables and can have multiple lagged predictor variable thresholds. In contrast, Tong's methodology creates threshold models from piecewise linear models whose terms are restricted to the initial sets of candidates of the ASTAR algorithm ((25) for our example). Tong's threshold models do not allow interactions among lagged predictor variables and are usually limited to a single threshold due to the difficulties associated with the threshold selection process.

We next examine the ability of ASTAR to identify and model linear and nonlinear times series models. The simulation of an AR(1) model with known coefficients examines the ability of ASTAR to detect and model a simple linear time series. The simulation of a threshold model with 'AR(1)-like' models in each disjoint subregion examines the ability of ASTAR to detect and model simple nonlinear threshold time series. Finally, in Section 6.3 we examine the ability of ASTAR to model the widely studied Wolf sunspot numbers, a nonlinear time series with periodic behavior.

6.1 AR(1) Simulations

We first consider simulation of an AR(1) model,

$$X_\tau = \rho X_{\tau-1} + K + \epsilon_\tau \tag{27}$$

where $\tau = 1, 2, \ldots, N$ indexes the time series, ρ is a constant coefficient varied within experiments, $K = 0$ is the model constant and ϵ_τ is $N(0, \sigma_\epsilon^2)$. The model is usually considered under the stationarity conditions ($| \rho | < 1$), but random walks ($| \rho | = 1$) and explosive processes ($| \rho | > 1$), are also of interest. Two categories of experiments were conducted using the AR(1) model. The first experiment required ASTAR to estimate a model from the simulated data of the AR(1) model using one lag predictor variable $X_{\tau-1}$, and using $M = 3$, the maximum number of subregions in the forward step ASTAR procedure. The first experiment's alternative models either have no $X_{\tau-1}$ term (a constant model) or have a $X_{\tau-1}$ term with a threshold value t greater than $\min\{X_{\tau-1}\}_{\tau=1}^{N-1}$. In this case we call the threshold value t an internal threshold. The second experiment required ASTAR to estimate a model from the simulated data of the AR(1) model using four lag predictor variables, $\{X_{\tau-i}\}_{i=1}^{4}$, and using $M = 8$, the maximum number of subregions allowed in the forward step ASTAR procedure. The second experiment's alternative models include constant models, models with an internal threshold value, and any model that includes a term other than $X_{\tau-1}$. The interest in these simulations is two-fold: how often was the true model identified, and if so, how well were the parameters K and ρ estimated.

Several simulation results are shown in Figures 2-7 for $\rho = .5, .7$ and $.9$, $K = 0$, and $\epsilon_\tau = \mathrm{N}(0,1)$. Each figure is a series of box plots for the coefficients of the 100 AR(1) simulated models correctly identified by ASTAR for increasing values of N. The true value of each model coefficient is identified by the dashed line across the box plots. At the top of each figure is the length N of each simulated time series, the number C of the 100 simulated models correctly identified by the ASTAR procedure, and the equivalent sample size for independent data, $Eq\,S\,SIZE = (N/\sum_{i=-\infty}^{\infty} \rho^i)$ (Priestley, 1981). Underneath each box plot is summary information for the coefficient estimates of the correctly identified AR(1) models i.e., the sample mean and sample standard deviation of the values in the box plots. By comparing the true and the estimated values of the model coefficients across increasing values of N it is observed that the estimated values of the coefficients tend to the true value as N increases. Also, in all but one simulation the number of correctly identified models (100-C) rises to 100 for increasing values of N. Note that the ASTAR estimates for ρ have negative bias for small values of N that generally decreases as N increases. The downward bias of $\hat{\rho}$ is similar to that identified by Kendall et al. (1983) and others when using data for estimating autocorrelations.

6.2 Threshold Simulations

To observe the ability of ASTAR to capture nonlinear threshold model characteristics we consider simulation

of the 2-subregion threshold model

$$X_\tau = \begin{cases} \rho_1 X_{\tau-1} + \epsilon_\tau & \text{if } X_{\tau-1} \leq 0 \\ \rho_2 X_{\tau-1} + \epsilon_\tau & \text{if } X_{\tau-1} > 0 \end{cases} \tag{28}$$

where $\tau = 1, 2, \ldots, N$ indexes the time series, ρ_1 and ρ_2 are constant coefficients varied for different experiments and ϵ is $N(0, \sigma_\epsilon^2)$. Note that the threshold model (28) has an 'AR(1)-like' model in each subregion. Two categories of experiments were conducted using the threshold model. The first experiment required ASTAR to estimate a model from the simulated data of the threshold model using one lag predictor variable $X_{\tau-1}$, and using $M = 3$, the maximum number of subregions in the forward step ASTAR procedure. The first experiment's alternative models are either linear or have more than one internal threshold. The second experiment required ASTAR to estimate a model from the simulated data of the threshold model using four lag predictor variables, $\{X_{\tau-i}\}_{i=1}^4$, and using $M = 10$, the maximum number of subregions allowed in the forward step ASTAR procedure. The second experiment's alternative models have more than one internal threshold term, include terms other than $X_{\tau-1}$, or are linear.

Several simulation results are shown in Figures 8-11 for $\rho_1, \rho_2 = .7, .3$ and $-.6, .6$, and $\epsilon_\tau = \mathrm{N}(0, .25)$. As with the previous AR(1) model simulation experiments, each figure is a series of box plots for the coefficients of the 100 threshold simulated models correctly identified by ASTAR for increasing values of N. The true value of each model coefficient is identified by the dashed line across the box plots. At the top of each figure is the length N of each simulated time series, and the number C of the 100 simulated models correctly identified by the ASTAR procedure. Underneath each box plot is summary information for the coefficient estimates of the correctly identified threshold models i.e., the sample mean and sample standard deviation of the values in the boxplots. Note that the number of correctly identified models rises for increasing values of N. However, a consistent improvement in the mean and standard deviation for the estimated values of the model coefficients is not always observed for increasing values of N. For the most part this is attributed to the increasing number of correctly identified models for increasing values of N.

6.3 Threshold Modeling of the Wolf's Sunspot Numbers

As an illustration of ASTAR ability to model an actual time series we examined 221 (1700-1920) of the Wolf sunspot numbers. The Wolf sunspot numbers are relative measures of the average monthly sunspot activity on the surface of the sun (see, e.g., Scientific American, February 1990). Some of the early analysis and modeling of the sunspot numbers was performed by Yule (1927) as an example for introducing autoregressive models. Recently suggested nonlinear models of the sunspot numbers include threshold models (Tong, 1983) and bilinear models

(Rao and Gabr, 1984). A detailed review of the history of the sunspot numbers is provided by Izenman (1983).

The data (Figure 12) is quite 'periodic' but has nonsymmetric cycles with extremely sharp peaks and troughs. The cycles (Table 1) generally vary between 10 and 12 years with the greater number of sunspots concentrated in each descent period versus the accompanying ascent period. The average ascent period is 4.60 years and the average descent period is 6.58 years. Attempts to model the data with a fixed cycle period signal plus (possibly correlated) noise have failed because the cyclic component in the spectrum (Figure 13, top) is quite spread out and diffuse.

Ascent period	5		5		4		5		6		6		3		3		3		6		6	
Descent period		7		6		6		6		5		5		6		6		11		6		7
Ascent (cont)	7		4		5		4		3		5		4		4		4					
Descent (cont)		3		6		8		7		8		6		8		8						

Table 1: Ascent and Descent periods of the Sunspot Data (1700-1920).

One of the interesting characteristics of Tong's analysis of the sunspot numbers included the development of threshold models with stationary harmonic behavior or limit cycles. Using Tong (1983), let $\tau = 1, 2, \ldots$ index a times series and let $\boldsymbol{x}_\tau^k = \{x_\tau, x_{\tau-1}, \ldots, x_{\tau-k+1}\}$ denote a k-dimensional vector in $D \in \Re^k$ that satisfies the equation,

$$\boldsymbol{x}_\tau^k = f(\boldsymbol{x}_{\tau-1}^k), \tag{29}$$

where f is a vector-valued function. Let $f^j(\boldsymbol{x})$ denote the jth iterate of f, i.e.,

$$f^j(\boldsymbol{x}) = \underbrace{f(f(f(\ldots(f}_{\text{j of them}}(\boldsymbol{x}))\ldots))). \tag{30}$$

We say that a k-dimensional vector $\boldsymbol{x}^{*k}$ is a *stable limit point* of the function f with respect to the domain D if

$$f^j(\boldsymbol{x}_0) \rightarrow \boldsymbol{x}^{*k} \text{ as } j \rightarrow \infty \quad \forall\, \boldsymbol{x}_0 \in D. \tag{31}$$

Also, we say that a k-dimensional vector $\boldsymbol{c}_1^k$ is a *stable periodic limit point* with period $T > 1$ of the function f with respect to the domain D if

$$f^{jT}(\boldsymbol{x}_0) \rightarrow \boldsymbol{c}_1^k \text{ as } j \rightarrow \infty \quad \forall\, \boldsymbol{x}_0 \in D, \tag{32}$$

and the convergence does not hold for any divisor of T. It follows that $\boldsymbol{c}_1^k, f^1(\boldsymbol{c}_1^k), f^2(\boldsymbol{c}_1^k), \ldots, f^{T-1}(\boldsymbol{c}_1^k)$ are simultaneously distinct stable periodic points of the function f with respect to D. If we let $f^i(\boldsymbol{c}_1^k)$ be denoted

by $c^k_{i+1}, i = 0, 1, \ldots, T-1$, then the set $\{c^k_1, c^k_2, c^k_3, \ldots, c^k_{T-1}\}$ is called a stable limit cycle of the function f with respect to D.

Our primary interest in limit cycles is for investigating the underlying characteristics of the true function $f(\boldsymbol{x})$ given at (1). If we believe that the cyclical behavior of $f(\boldsymbol{x})$ can be modeled as a limit cycle perturbed by Gaussian white noise, as we do with the sunspot numbers, then when applying ASTAR to the sunspot numbers it would be satisfying to identify an underlying limit cycle in our estimate of $f(\boldsymbol{x})$. With this objective in mind we investigated 20 ASTAR models of the sunspot numbers. The different models were identified by varying the user parameters of the ASTAR algorithm to include; level of interaction, number and separation of partition points, number of forward step subregions, and availability of lagged predictor variables. The maximum order of the model (number of lagged predictor variables) was restricted to 20 and the first 20 sunspots (1700-1719) were used for model initialization.

Table 2 provides a summary of the 20 ASTAR models for the sunspot numbers (1720-1920), ordered by the generalized cross validation (GCV) criterion (24). The first three columns identify the model number, the GCV criterion and the mean sum of squares (MSS) of the fitted residuals for each ASTAR model. The fourth through sixth columns identify the number of estimated parameters, the number of partition points and the maximum level of interaction in each model. Columns seven and eight identify the length (in years) of each model's limit cycle (if one exists) and the number and lengths (in years) of the one or more type 'subcycles' (ascent and descent periods) within the limit cycle. We use MSS instead of $\text{MSS}^{1/2}$ to facilitate comparison of the ASTAR models with other modeling efforts of the Sunspot numbers.

	GCV	MSS	Number of Model Parameters	Number of Interior Thresholds	Level of Model Interaction	Length of Limit Cycle (in years)	Number (Lengths) of Subcycles
1	141.6	91.6	16	4	4	225	27 (8,9)
2	159.9	111.7	14	4	3	9	1 (9)
3	160.5	91.4	25	9	3	—	
4	164.8	95.3	18	4	5	—	
5	165.9	113.0	19	7	2	9	1 (9)
6	166.2	115.9	13	3	4	120	11 (10,11)
7	167.2	114.2	14	3	4	—	
8	173.9	119.5	14	3	3	—	
9	174.1	114.2	14	6	3	137	13 (10,11)
10	176.8	125.6	11	2	2	78	7 (11,12)
11	180.1	101.0	15	6	4	43	4 (10,11)
12	180.3	115.9	13	3	3	—	
13	184.1	119.8	11	2	3	133	12 (11,12)
14	187.9	103.6	18	3	4	—	
15	190.2	126.2	13	1	3	—	
16	191.4	110.5	17	3	3	167	15 (11,12)
17	193.5	116.0	13	2	4	94	10 (9,10)
18	195.3	117.6	15	2	4	—	
19	195.5	114.2	17	3	3	120	11 (10,11)
20	211.1	119.6	18	3	3	23	4 (5,6)

Table 2: ASTAR models of the Wolf Sunspot Numbers (1720-1920).

Some form of a limit cycle exists in 12 of the 20 ASTAR models. Also, 5 of the 12 models, namely 9,10,11,13 and 16, provide limit cycles with lengths 137,78,43,133 and 167 respectively, and 'subcycles' with lengths and range similar enough to the behavior of the sunspot data (Table 1) to warrant further analysis. Of these 5 models, 2 (9 and 11) provide fitted residuals that appear independent and Gaussian. Some of the statistics for the fitted residuals of these two models are provided in Table 3.

	Model 9	Model 11	
Mean	0.000	0.000	
MSS	114.2	101.0	
Skewness	0.0813	.346	0 for normal distribution
Kurtosis	0.673	0.153	0 for normal distribution
K-S	.275	.349	level of significance
C-M	$> .15$	$> .15$	level of significance
A-D	$> .15$	$> .15$	level of significance
L-M	.6892	.0466	level of significance

Table 3: Statistics for the Fitted Residuals of ASTAR Models 9 and 11 of the Wolf sunspot numbers (1720-1920).

The Skewness and Kurtosis statistics serve as a general indicator of the symmetry and heaviness of the tails for the sample distribution function of the fitted residuals $\hat{F}_{\epsilon}(x)$. The Kolmogorov-Smirnov (K-S) test statistic measures the maximum absolute distance between $\hat{F}_{\epsilon}(x)$ and the hypothesized true normal N(0,1) distribution function $F_{x}(x)$ while the Cramer-von Mises (C-M) statistic measures the integral of the squared distance between the two functions. A drawback to the K-S and C-M tests are that they lack sensitivity to departures from the null hypothesis that occur in the tails of a distribution. As an approach to overcome the lack of sensitivity of the K-S and C-M tests, the Anderson-Darling (A-D) test statistic weights the distances between the two functions. A final test for independent and Gaussian error structure is provided by the Lin-Mudhoekar (L-M) test statistic which tests for asymmetry. We rejected Model 11 due to the low level of significance of the L-M test statistic and identified Model 9 as the best model (with limit cycle) of the 20 models considered in the initial analysis. Model 9 is

$$X_{\tau} = \begin{cases} 2.710606 + .959891X_{\tau-1} + .331893(47.0 - X_{\tau-5})_{+} - .257034(59.1 - X_{\tau-9})_{+} \\ -.002707X_{\tau-1}(X_{\tau-2} - 26.0)_{+} + .016674X_{\tau-1}(44.0 - X_{\tau-3})_{+} - .031516X_{\tau-1}(17.1 - X_{\tau-4})_{+} \\ +.004166X_{\tau-1}(26.0 - X_{\tau-2})_{+}(X_{\tau-5} - 41.0)_{+} \end{cases} \quad (33)$$

where $(x)_{+}$ is a plus function with value x if $x > 0$ and 0 otherwise. Model 9 has 8 terms (a constant term with 3 one-way, 3 two-way and 1 three-way interactions) and 6 threshold values (1 each on $X_{\tau-2}, X_{\tau-3}, X_{\tau-4}$, and $X_{\tau-9}$ and 2 on $X_{\tau-5}$).

Figures 12-17 are various plots of the fitted values and residuals of ASTAR Model 9. Figure 12 shows the fitted values of the model versus the Wolf sunspot numbers (1720-1920). The model appears to equally overfit and underfit the peaks and troughs as it captures the general structure of the sunspot numbers. The model fit is further examined using the estimated normalized periodogram (Figure 13) and autocorrelation function plots (Figure 14). The fitted residuals of the model are examined using residual versus time and fit plots (Figure 15) and the residual autocorrelation function plot (Figure 16). In Figure 15 the slight lack of negative residuals for small fitted values of the model is attributed to the sunspot numbers being positive random variables. Figure 17 shows the 137 year limit cycle of Model 9 with its ascent and descent periods. The limit cycle is asymmetric with a range in amplitude of 17.7 to 94.5 and an average ascent/descent period of 4.3/6.23 years versus 4.6/6.58 years for the actual sunspot numbers. In comparing Model 9's limit cycle (Figure 17) with the real sunspot data (Figure 12) note that the standard deviation of the fitted residual's error variance is estimated as $(\text{MSS})^{1/2} = 10.69$ sunspots.

To investigate the predictive performance of ASTAR Model 9, developed using the Sunspot numbers from

1700-1920, we compared it's forward step predictions with the Full Autoregressive, Threshold (Tong, 1983) and Bilinear subset (Rao,1984) models for the Sunspot numbers from 1921-1955. The mean sum of squares for the errors of the predictions obtained by these four models are given in Table 4.

Model	AR	Threshold (Tong)	Bilinear Subset (Rao)	ASTAR Model 9
$\hat{\sigma}_\epsilon^2$	199.27	153.71	124.33	114.2
Number of Parameters	10	19	11	14
$\hat{\sigma}_\epsilon^2(1)$	190.9	148.2	123.8	136.4
$\hat{\sigma}_\epsilon^2(2)$	414.8	383.9	337.5	324.1
$\hat{\sigma}_\epsilon^2(3)$	652.2	675.6	569.8	481.0
$\hat{\sigma}_\epsilon^2(4)$	725.8	716.1	659.0	427.3
$\hat{\sigma}_\epsilon^2(5)$	771.0	756.4	718.9	378.0
$\hat{\sigma}_\epsilon^2(6)$	—	—	—	420.8
$\hat{\sigma}_\epsilon^2(7)$	—	—	—	454.2
$\hat{\sigma}_\epsilon^2(8)$	—	—	—	468.6

Table 4: The mean sum of squares error $\hat{\sigma}_\epsilon^2$, number of model parameters and the predictive mean sum of squares error $\hat{\sigma}_\epsilon^2(i)$ for the ith forward step prediction for the period (1921-1955) of the AR, Threshold, Bilinear and ASTAR models of the Sunspot Numbers for the period (1700-1920).

The performance of the ASTAR model for forecasting the Sunspot numbers from 1921-1955 is a considerable improvement over the AR and Threshold models for every forward step and is an improvement over the Bilinear subset model for every forward step with the exception of the first step. Also, it is interesting and very surprising to note that the predictive mean sum of squares error for the ASTAR model decreases in the fourth and fifth step before increasing again. This phenomenon was also identified in subsequent analysis of other ASTAR models with limit cycles. We attribute this interesting phenomenon to the underlying limit cycle of the models.

7 CONCLUSIONS

MARS is a new methodology for nonparametric modeling that utilizes regression spline modeling and a modified recursive partitioning strategy to exploit the localized low dimensional behavior of the data used to construct $\hat{f}(\boldsymbol{x})$. Although MARS is a computationally intensive regression methodology, it can produce continuous models for high dimensional data that can have multiple partitions and predictor variable interactions. The final MARS model may be analyzed using an ANOVA style decomposition. Also, although the main advantages of MARS modeling are in high dimensional settings $p > 2$ it has been shown to be highly competitive with other regression methods in low dimensional settings (Friedman, 1988).

In this paper, by letting the predictor variables in MARS be lagged values of a time series, we obtain ASTAR (Adaptive Spline Threshold Autoregression), a new method for nonlinear threshold modeling of time series. We show this by applying ASTAR to simple autoregressive and nonlinear threshold models. A significant feature of ASTAR when modeling time series data with periodic behavior is its ability to produce continuous models with underlying sustained oscillations (limit cycles). Time series that possess such behavior include the Wolf Sunspots, Canadian Lynx and various river flow data sets to name a few. Our initial analysis of the Wolf sunspot numbers (1700-1920) using ASTAR produced several models with underlying limit cycles. When used to predict the Sunspot numbers (1921-1955), the ASTAR models are a significant improvement over existing Threshold and Bilinear models.

ACKNOWLEDGMENTS

The research of P.A.W.Lewis was supported at the Naval Postgraduate School under an Office of Naval Research Grant. The interest of P.A.W. Lewis in nonlinear threshold models was stimulated by attendance at the Research Workshop on Non-linear Time Series, organized by H. Tong at Edinburgh, Scotland from 12-15 July 1989. The authors are also grateful to Ed McKenzie for his helpful ideas when applying MARS to the modeling of time series.

References

Bellman, R. E., *Adaptive Control Processes*, Princeton University Press, Princeton, New York, 1961.

Breiman, L., Friedman, J., Olshen, R., and Stone, C., *Classification and Regression Trees*, Wadsworth, Belmont, CA., 1984.

Breiman, L. and Meisel, W. S., "General Estimates of the Intrinsic Variability of Data in Nonlinear Regression Models", *Journal of the American Statistical Association*, v. 71, pp. 301–307, 1976.

Craven, P. and Wahba, G., "Smoothing Noisy Data With Spline Functions. Estimating the Correct Degree of Smoothing by the Method of Generalized Cross-Validation", *Numerische Mathematik*, v. 31, pp. 317–403, 1979.

Eubank, R. L., *Spline Smoothing and Nonparametric Regression*, Marcel Dekker Inc., New York, 1988.

Friedman, J. H., *Multivariate Adaptive Regression Splines*, Department of Statistics, Stanford University, Report 102, November 1988.

Granger, C. W. and Anderson, A. P., "An Introduction to Bilinear Time Series Models", *Vandenhoeck and Ruprecht*, pp. 1–94, 1978.

Izenman, A. J., "J. R. Wolf and J. A. Wolfer: An Historical Note on the Zurich Sunspot Relative Numbers", *Journal of the Royal Statistical Society, (Ser A)*, v. 146, no. 3, pp. 311–318, 1983.

Kendall, M., Stuart, A., and Ord, J. K., *The Advanced Theory of Statistics*, 4th ed., v. 3, Macmillan, 1983.

Lewis, P. A. W. and Orav, J., *Simulation Methodology for Statisticians, Operations Analysts, and Engineers*, v. 1, Wadsworth & Brooks/Cole, 1988.

Morgan, J. N. and Sonquist, J. A., "Problems in the Analysis of Survey Data, and a Proposal", *Journal of the American Statistical Association*, v. 58, pp. 415–434, 1963.

Priestley, M. B., *Spectral Analysis and Time Series*, Academic Press, 1981.

Priestley, M. B., *Non-Linear and Non-Stationary Times Series*, Academic Press, 1988.

Rao, T. S. and Gabr, M. M., *An Introduction to Bispectral Analysis and Bilinear Time Series Models*, Springer-Verlag, 1984.

Shumaker, L. L., *Spline Functions*, John Wiley and Sons, Inc., 1981.

Silverman, B. W., "Some Aspects of the Spline Smoothing Approach to Non-Parametric Regression Curve Fitting", *Journel of the Royal Stastical Society (Ser B)*, v. 47, no. 1, pp. 1–52, 1985.

Smith, P. L., "Splines as a Useful and Convenient Statistical Tool", *The American Statistician*, v. 33, no. 2, pp. 57–62, 1979.

Tong, H., *Threshold Models in Non-linear Time Series Analysis*, Springer-Verlag, 1983.

Tong, H. and Lim, K. S., "Threshold Autoregression, Limit Cycles and Cyclical Data", *Journal of the Royal Statistical Society (Ser B)*, v. 42, no. 3, pp. 245–292, 1980.

Wegman, E. J. and Wright, I. W., "Splines in Statistics", *Journal of the American Statistical Association*, v. 78, no. 382, pp. 351–365, 1983.

Wold, S., "Spline Functions in Data Analysis", *Technometrics*, v. 16, no. 1, pp. 1–11, Feburary 1974.

Yule, G. U., "On a Method of Investigating Periodicities in Disturbed Series with Special Reference to Wolfer's Sunspot Numbers", *Philos. Trans. Roy. Soc (Ser A)*, v. 226, pp. 267–298, 1927.

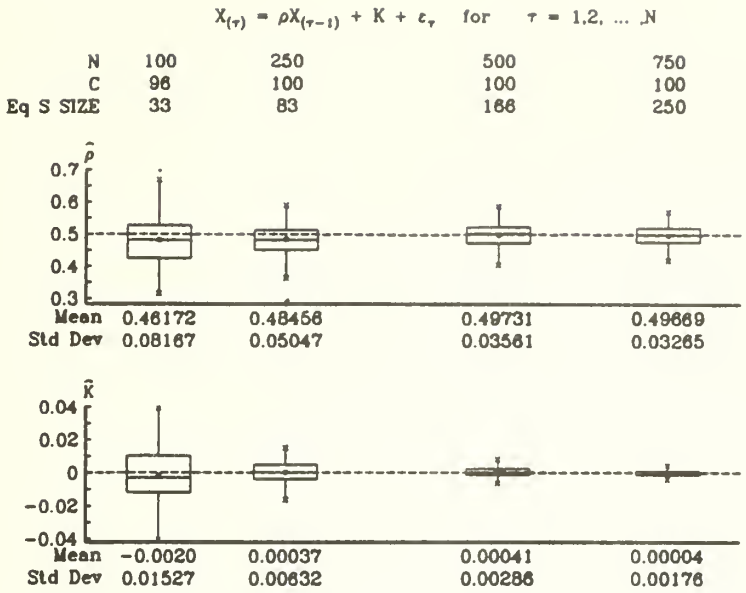

Figure 2: AR(1) MODEL SIMULATION: ASTAR estimates for $\rho = .5, K = 0$ and $\sigma_\epsilon^2 = N(0,1)$ from C simulations of an AR(1) model for increasing values of N, with $P = 1$ lag predictor variables, and $M = 3$, the number of forward step subregions permitted in the ASTAR algorithm. Each simulation consists of 100 replications. The boxplots are for the estimates of the model parameters when ASTAR correctly identified the AR(1) model. *For $N = 100$, 2 simulations were incorrectly identified as constant models.*

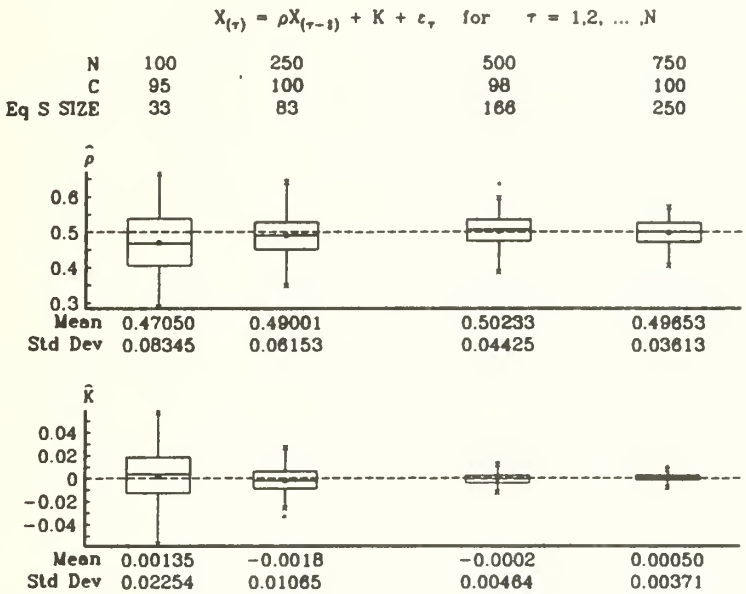

Figure 3: AR(1) MODEL SIMULATION: ASTAR estimates for $\rho = .5, K = 0$ and $\sigma_\epsilon^2 = N(0,1)$ from C simulations of an AR(1) model for increasing values of N with $P = 4$ lag predictor variables, and $M = 8$, the number of forward step subregions permitted in the ASTAR algorithm. Each simulation consists of 100 replications. The boxplots are for the estimates of the model parameters when ASTAR correctly identified the AR(1) model. *For $N = 100$, 5 simulations were incorrectly identified as; 2 constant models, 1 AR(2) model and 2 AR(3) models. For $N = 500$, 2 simulations were incorrectly identified as constant models.*

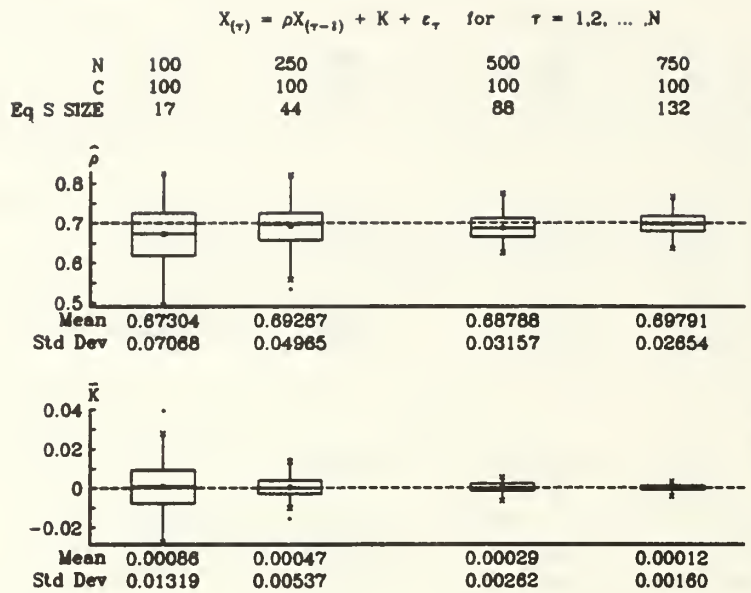

Figure 4: AR(1) MODEL SIMULATION: ASTAR estimates for $\rho = .7, K = 0$ and $\sigma_\epsilon^2 = N(0,1)$ from C simulations of an AR(1) model for increasing values of N, with $P = 1$ lag predictor variables, and $M = 3$, the number of forward step subregions permitted in the ASTAR algorithm. Each simulation consists of 100 replications. The boxplots are for the estimates of the model parameters when ASTAR correctly identified the AR(1) model. *Here all cases were correctly identified.*

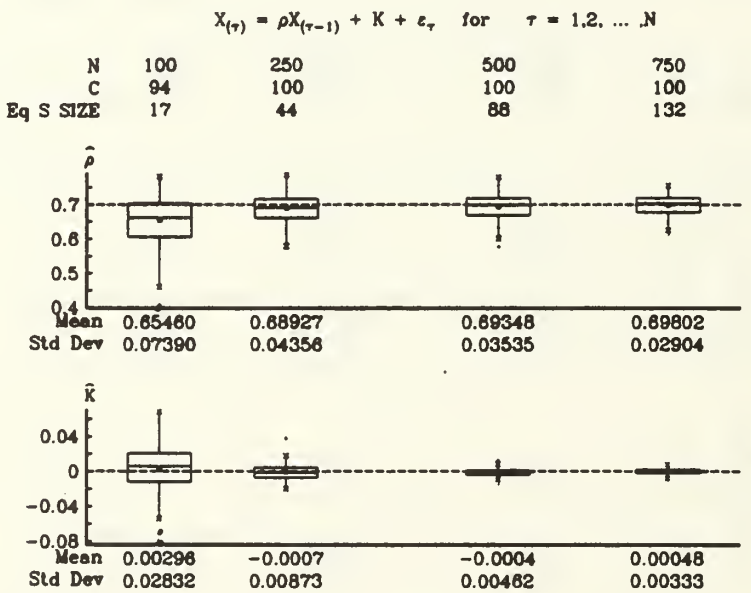

Figure 5: AR(1) MODEL SIMULATION: ASTAR estimates for $\rho = .7, K = 0$ and $\sigma_\epsilon^2 = N(0,1)$ from C simulations of an AR(1) model for increasing values of N with $P = 4$ lag predictor variables, and $M = 8$, the number of forward step subregions permitted in the ASTAR algorithm. Each simulation consists of 100 replications. The boxplots are for the estimates of the model parameters when ASTAR correctly identified the AR(1) model. *For $N = 100$, 6 simulations were incorrectly identified as; 2 AR(2) models, 2 AR(3) model and 2 AR(4) models.*

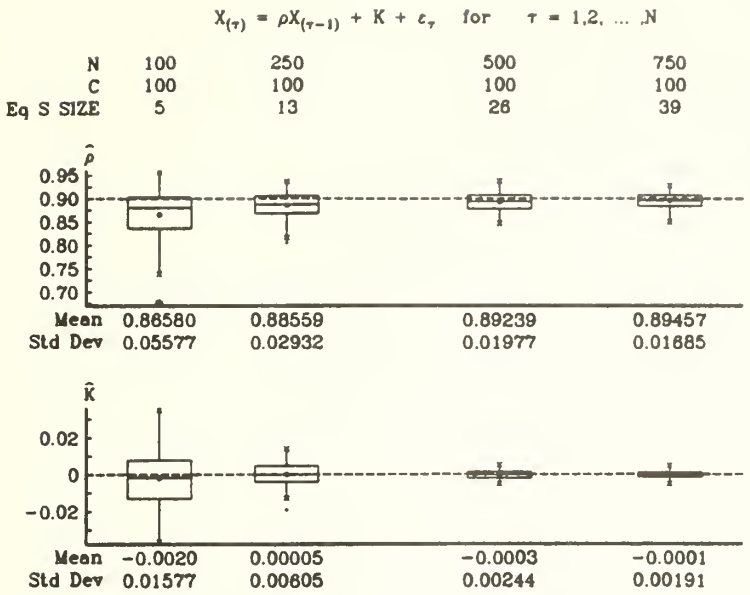

Figure 6: AR(1) MODEL SIMULATION: ASTAR estimates for $\rho = .9, K = 0$ and $\sigma_\epsilon^2 = N(0,1)$ from C simulations of an AR(1) model for increasing values of N, with $P = 1$ lag predictor variables, and $M = 3$, the number of forward step subregions permitted in the ASTAR algorithm. Each simulation consists of 100 replications. The boxplots are for the estimates of the model parameters when ASTAR correctly identified the AR(1) model. *Here all cases were correctly identified.*

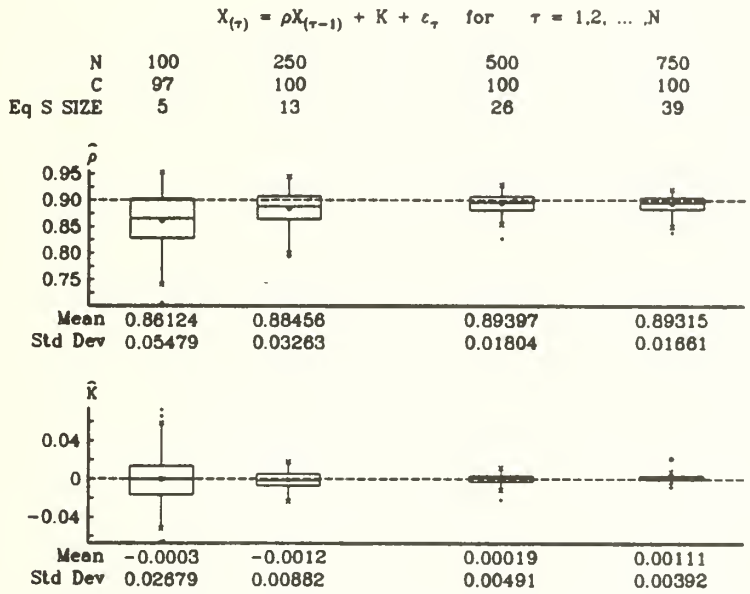

Figure 7: AR(1) MODEL SIMULATION: ASTAR estimates for $\rho = .9, K = 0$ and $\sigma_\epsilon^2 = N(0,1)$ from C simulations of an AR(1) model for increasing values of of N with $P = 4$ lag predictor variables, and $M = 8$, the number of forward step subregions permitted in the ASTAR algorithm. Each simulation consists of 100 replications. The boxplots are for the estimates of the model parameters when ASTAR correctly identified the AR(1) model. *For $N = 100$, 3 simulations were incorrectly identified as; 2 AR(2) models and 1 AR(3) model.*

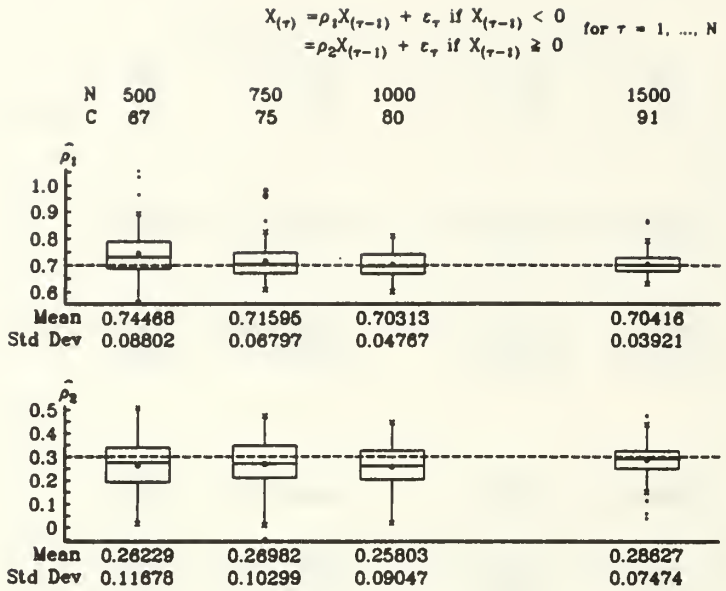

Figure 8: THRESHOLD MODEL SIMULATION: ASTAR estimates for $\rho_1, \rho_2 = .7, .3$ and $\sigma_\epsilon^2 = N(0, .25)$ from C simulations of a threshold model for increasing values of N, with $P = 1$ lag predictor variables, and $M = 3$, the number of forward step subregions permitted in the ASTAR algorithm. Each simulation consists of 100 replications. The boxplots are for the estimates of the model parameters when ASTAR correctly identified the threshold model. *The models of the simulations that ASTAR did not correctly identify as the threshold model (28) contained an incorrect number of subregions or lacked an AR(1) term in one of the two subregions.*

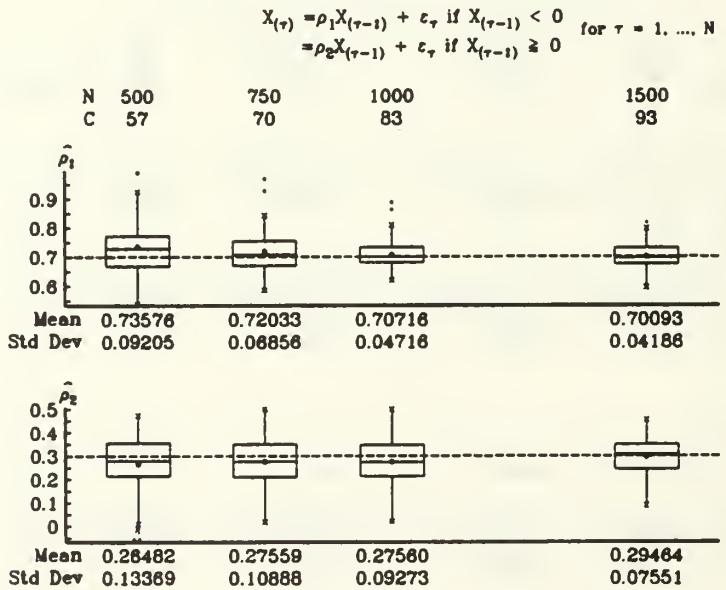

Figure 9: THRESHOLD MODEL SIMULATION: ASTAR estimates for $\rho_1, \rho_2 = .7, .3$ and $\sigma_\epsilon^2 = N(0, .25)$ from C simulations of a threshold model for increasing values of N, with $P = 4$ lag predictor variables, and $M = 10$, the number of forward step subregions permitted in the ASTAR algorithm. Each simulation consists of 100 replications. The boxplots are for the estimates of the model parameters when ASTAR correctly identified the threshold model. *The models of the simulations that ASTAR did not correctly identify as the threshold model (28) contained an incorrect number of subregions, lacked an AR(1) term in one of the two subregions or contained terms with $X_{\tau-2}, X_{\tau-3}$, or $X_{\tau-4}$.*

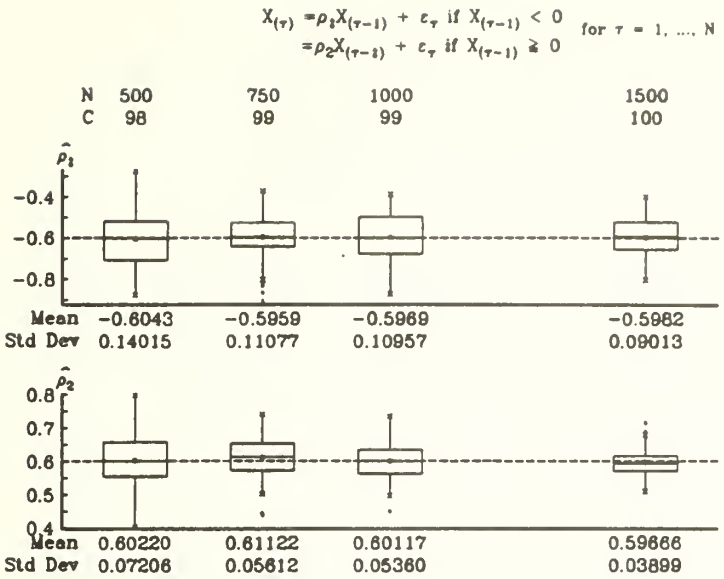

Figure 10: THRESHOLD MODEL SIMULATION: ASTAR estimates for $\rho_1, \rho_2 = -.6, .6$ and $\sigma_\epsilon^2 = N(0, .25)$ from C simulations of a threshold model for increasing values of N, with $P = 1$ lag predictor variables, and $M = 3$, the number of forward step subregions permitted in the ASTAR algorithm. Each simulation consists of 100 replications. The boxplots are for the estimates of the model parameters when ASTAR correctly identified the threshold model *The models of the simulations that ASTAR did not correctly identify as the threshold model (28) contained an incorrect number of subregions or lacked an AR(1) term in one of the two subregions.*

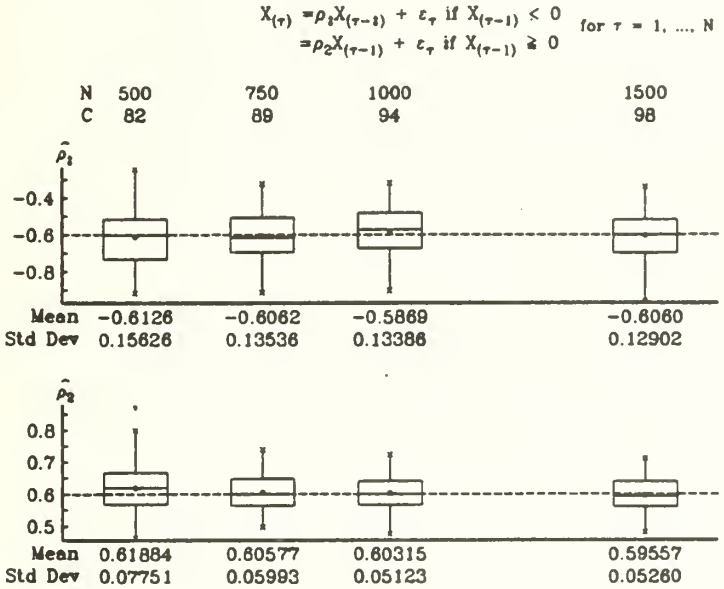

Figure 11: THRESHOLD MODEL SIMULATION: ASTAR estimates for $\rho_1, \rho_2 = -.6, .6$ and $\sigma_\epsilon^2 = N(0, .25)$ from C simulations of a threshold model for increasing values of N, with $P = 4$ lag predictor variables, and $M = 10$, the number of forward step subregions permitted in the ASTAR algorithm. Each simulation consists of 100 replications. The boxplots are for the estimates of the model parameters when ASTAR correctly identified the threshold models *The models of the simulations that ASTAR did not correctly identify as the threshold model (28) contained an incorrect number of subregions, lacked an AR(1) term in one of the two subregions or contained terms with* $X_{\tau-2}, X_{\tau-3}$, *or* $X_{\tau-4}$.

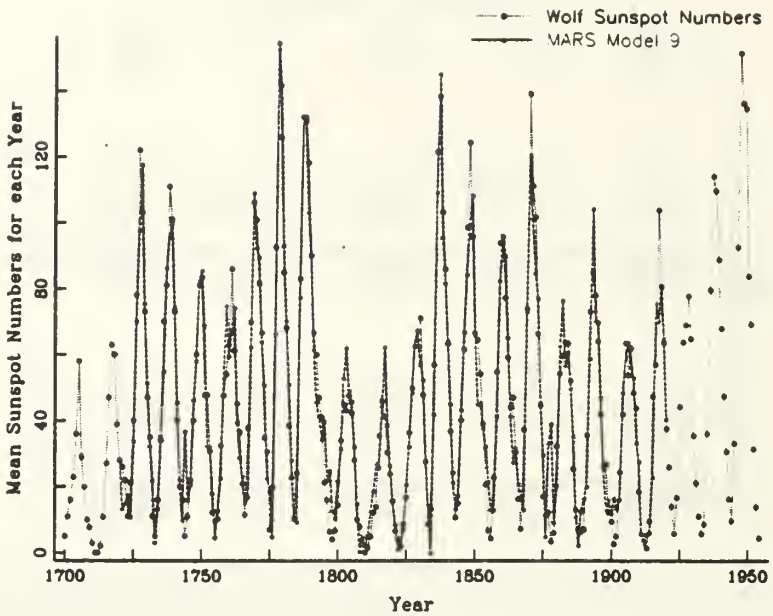

Figure 12: The Wolf Sunspot Numbers (1700-1955) versus the fit of ASTAR Model 9 (1720-1920). The sunspot numbers (1700-1719) were used for initialization. The sunspot numbers (1921-1955) were used to examine the prediction performance of ASTAR Model 9 and other models of the sunspot numbers.

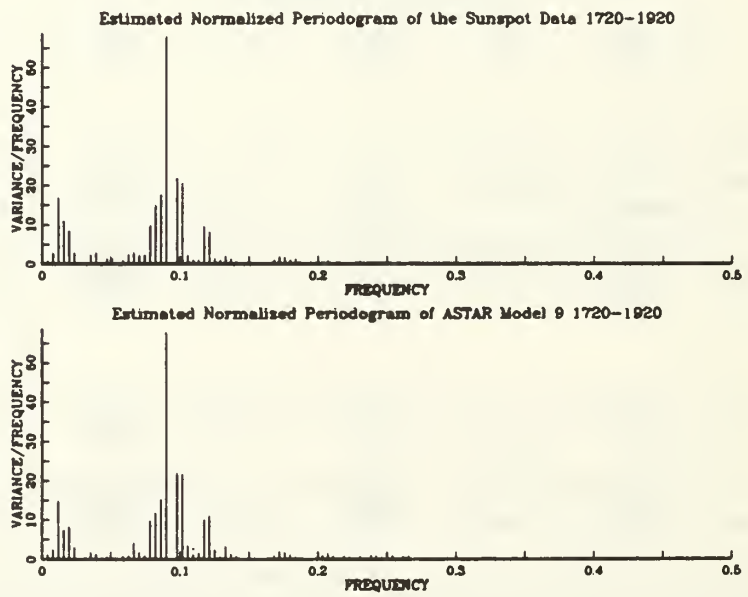

Figure 13: The estimated normalized periodogram of the Wolf Sunspot Numbers (1720-1920) [top] versus the estimated normalized periodogram of ASTAR Model 9 (1720-1920) [bottom]. The broad conclusion from the top periodogram is that there is a rather diffuse cycle in the data with a period of about 11 years, and a longer period of about 67 years.

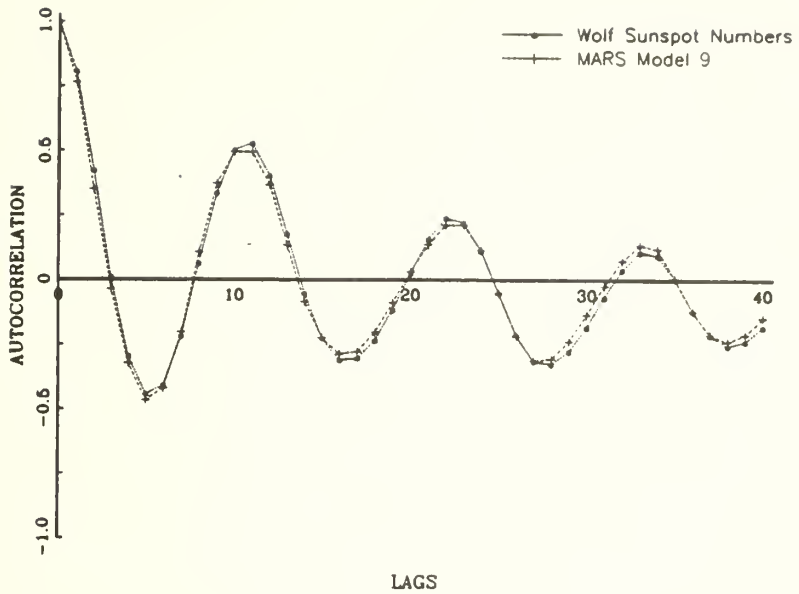

Figure 14: The autocorrelation functions of the Wolf sunspot numbers and ASTAR Model 9 for the period 1720-1920. The dominant cycle of period approximately 11 years is clearly evident.

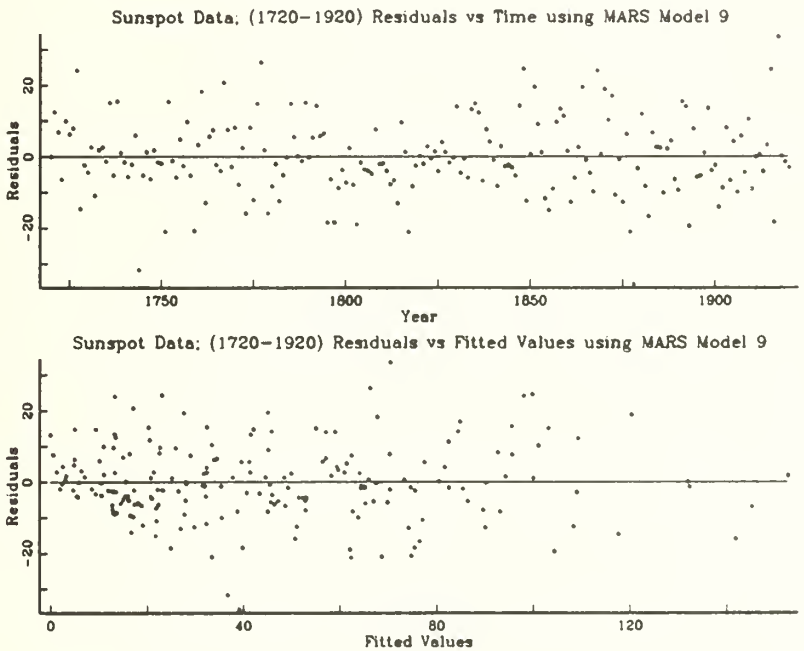

Figure 15: Fitted residuals from ASTAR Model 9 of the Wolf sunspot numbers (1720-1920) versus year [top]. Fitted residuals versus the fitted sunspot numbers from ASTAR Model 9 of the Wolf sunspot numbers (1720-1920) [bottom].

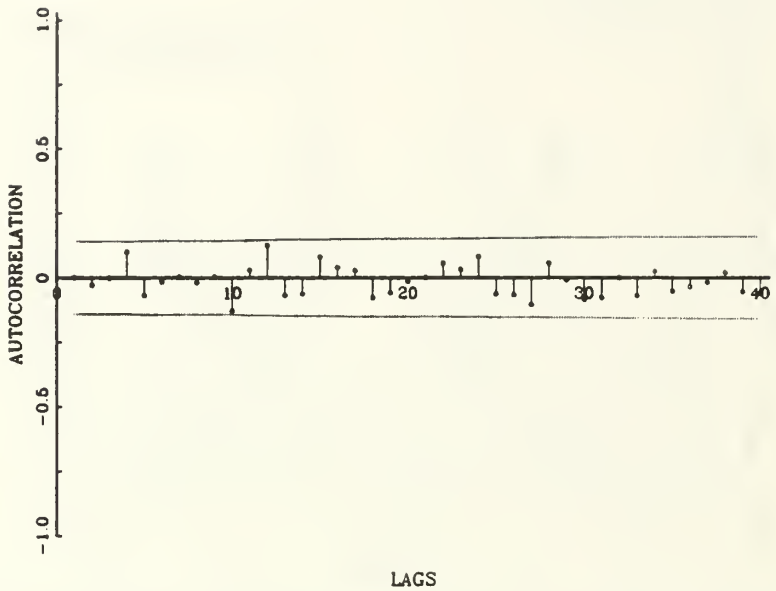

Figure 16: The autocorrelation function (first 40 lags) of the fitted residuals for ASTAR Model 9 of the Wolf sunspot numbers (1720-1920). There is no pattern of dependence in the residuals.

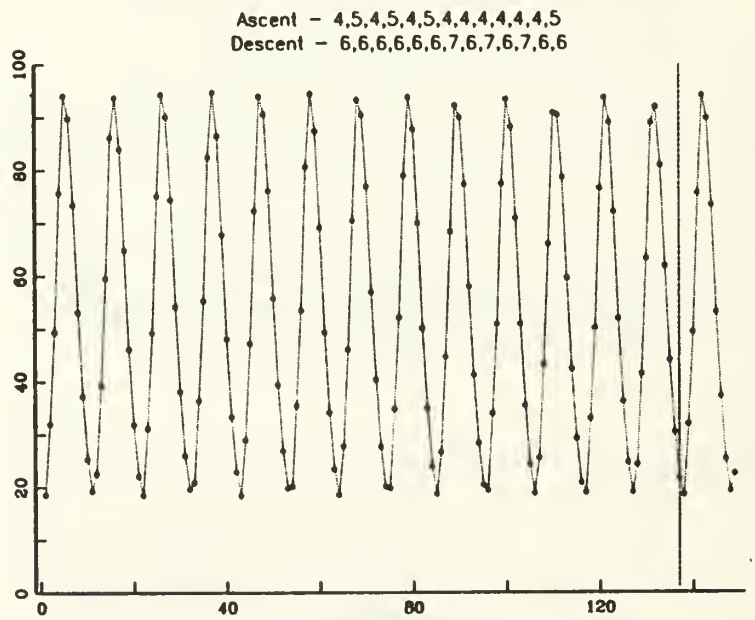

Figure 17: The limit cycle for ASTAR Model 9 of the Wolf sunspot numbers (1720-1920). The limit cycle is 137 years long with the indicated ascent and descent periods. The limit cycle is generated using ASTAR Model 9 initialized with the sunspot numbers (1700-1719). The 'subcycles' have lengths of 10 or 11 years with 4 or 5 years per ascent period and 6 or 7 years per descent period.

INITIAL DISTRIBUTION LIST

1. Library (Code 0142) 2
   Naval Postgraduate School
   Monterey, CA 93943-5000

3. Defense Technical Information Center 2
   Cameron Station
   Alexandria, VA 22314

4. Office of Research Administration 1
   Code 012A
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Prof. Peter Purdue, Code OR/Pd 1
   Naval Postgraduate School
   Monterey, CA 93943-5000

6. Prof. Peter A. W. Lewis 105
   Code OR/Lw
   Naval Postgraduate School
   Monterey, CA 93943-5000